

The development of a multimedia reader
for distance learning.

By
Erich Reichel

in conjunction with

Prof. ASJ Helberg

Acknowledgements

To my lord and saviour Jesus Christ.

To Marique for all your support and keeping me sane when my computer drove me insane.

To the Open Source community for allowing me to shamelessly copy your work to use as my own. It has been a very eventful learning process. Your products truly speak for themselves and leaves very little room for improvement.

To my project leader Prof ASJ Helberg, it has been a wonderful learning experience. If it weren't for this project I would still be using patented software.

All my friends, your assistance and kind words are always welcome.

Executive Summary

The goal of this project was to produce a device which can provide distance learning students with a method of learning by means of enriched study material. This device must be able to display study material in the form of text, pictures/graphs, sound and video. It should also be low-cost, and therefore low-technology.

Study material will be sent to students by mail, on CD-ROM or DVD. The costs connected with this, is very little with respect to the amount of study material that is transferred. All material can be incorporated in the standard webpage format, HTML.

A final version of the device, met all specifications. The software was manipulated to fit on a 512Mb CF2 card, leaving 140Mb space for user files. Linux was used to cut the total cost of the device, while still producing a fully functional system. Linux was then also used to speed the device up. Some problems were encountered with device drivers, but then the hardware was not functional under Windows either. These problems were solved by replacing the hardware.

This final system manages to display today's graphics with yesterday's hardware speed. The Linux system is fully optimized to use MMX and SSE functions (native to the final processor) which speeds the hardware up more than enough to meet all set specifications.

The use of Linux in this project brought the price down by roughly 25%. Further price cuts may be done by developing our own custom SBC (Single Board Computer) for use in this project. Developing our own SBC will cut the cost by excluding unused features such as network hardware.

Opsomming

Die doel van hierdie projek was om afstandonderrig-studente te voorsien van toerusting waarmee verrykte studiemateriaal gebruik kan word. Die produk moet in staat wees om studiemateriaal in die vorm van teks, grafika/grafieke, klank en video te kan vertoon. Dit moet ook 'n lae koste hê, en gevolglik van laetegnologie gebruik maak.

Studiemateriaal sal per pos aan studente gestuur word, op CD-ROM of DVD. Die koste om dit te pos, is baie klein in verhouding tot die groot hoeveelheid leermateriaal wat versend is. Alle studiemateriaal kan vertoon word deur van die standaard webformaat, HTML, gebruik te maak.

Die finale sisteem het alle spesifikasies bereik terwyl die sagteware steeds op 'n 512Mb CF2 kaart kon pas met 140Mb spasie beskikbaar vir die gebruiker. Linux is gebruik om die koste van die totale stelsel te verlaag, maar 'n funksionele eenheid is steeds daar gestel. Linux moes ook geoptimaliseer word om die produk se spoed te verbeter. Daar het probleme opgeduik met die installering van hardewaredrywers onder Linux, maar dié het ook nie onder Windows gewerk nie. Nuwe hardware moes gebruik word om die probleem op te los.

Die finale sisteem vertoon vandag se grafika op gister se hardewarespoed. Linux is ten volle geoptimaliseer vir MMX en SSE funksies (ingebou in die finale prosesseerder). Die hardware is sodoende vinnig genoeg om die spesifikasies met gemak te haal.

Die gebruik van Linux bring die totale koste met sowat 25% af. Die prys kan verder verlaag word deur ons eie SBC te ontwikkel. 'n Eie SBC kan die onnodige eienskappe soos netwerkhardeware uitsluit en sodoende die koste van die hardware verminder.

Table of Contents

1. Chapter One – Problem Statement.....	1
1.1. Digital Divide.....	1
1.2. Distance learning.....	2
1.3. Problem history.....	4
1.4. Problem extraction.....	6
1.4.1. Soft systems thinking.....	6
1.4.1.2. The Subsystems or components.....	8
1.4.1.3. The environment.....	10
1.4.1.4. Summary of the systems thinking model.....	10
1.4.2. Problem Definition.....	11
1.4.3. Specifications.....	12
1.4.3.1. Data read and display capability.....	12
1.4.3.2. Data storage capability.....	13
1.4.3.3. Portable and lightweight.....	13
1.4.3.4. Sufficient screen size.....	14
1.4.3.5. Colour screen to enhance viewing material.....	14
1.4.3.6. Sound output for enhanced material.....	14
1.4.3.7. Text input device.....	15
1.4.3.8. Pointer input device.....	16
1.4.3.9. Sound input capability (Optional).....	16
1.4.3.10. User friendly.....	16
1.4.3.11. Low cost.....	17
1.4.3.12. Low power usage.....	17
1.5. Conclusion.....	17
1.6. Overview of the rest of this document.....	18
2. Chapter Two – Literature study.....	19
2.1. Introduction.....	19
2.2. Hardware (What is available).....	19
2.2.1. Portable computing systems.....	20
2.2.2. EBook readers.....	21
2.2.3. Portable DVD.....	22
2.2.4. The Desktop PC.....	23
2.2.5. Summary.....	24
2.2.6. New development.....	24
2.2.6.1. Comparison with a general purpose desktop PC.....	26
2.2.6.2. Processor board.....	26
2.2.6.3. Mass storage media.....	27
2.2.6.4. I/O devices.....	28
2.3. Software.....	29
2.3.1. Choosing an OS.....	30
2.3.1.1. Graphical User Interface.....	30
2.3.1.2. 24/7 running time.....	31
2.3.1.3. Multimedia Capabilities.....	31
2.3.1.4. Time sharing between processes.....	31
2.3.1.5. Storage capabilities.....	31

2.3.2.	What is an OS.....	31
2.3.3.	The function of an OS.....	32
2.3.4.	Why we need an OS.....	34
2.3.5.	Desirable features of an OS.....	35
2.3.5.1.	Efficiency.....	35
2.3.5.2.	Reliability.....	35
2.3.5.3.	Maintainability.....	35
2.3.5.4.	Small Size.....	35
2.4.	Summary of an OS.....	36
2.4.1.	The OS applied to our project.....	36
2.5.	Different OS's.....	36
2.5.1.1.	MS-DOS.....	36
2.5.1.2.	Microsoft Windows.....	37
2.5.1.3.	Linux.....	37
2.5.1.4.	UNIX.....	38
2.5.1.5.	Comparison of OS's.....	39
2.5.2.	GNU Public License.....	40
2.5.2.1.	What does open source mean?.....	41
2.5.3.	Assessment of the open source agreement.....	42
2.5.3.1.	Conclusion.....	43
2.6.	Applications.....	43
2.6.1.	X.....	44
2.6.2.	Window managers.....	46
2.6.2.1.	Fluxbox [19].....	46
2.6.2.2.	XFCE.....	46
2.6.2.3.	KDE.....	46
2.6.3.	Browsers.....	47
2.6.3.1.	Mozilla.....	47
2.6.3.2.	Mozilla-Firefox.....	48
2.6.3.3.	Netscape.....	48
2.6.4.	Sound programs.....	49
2.6.4.1.	XMMS.....	49
2.6.4.2.	BMP.....	49
2.6.4.3.	JUK.....	49
2.6.5.	Video players.....	50
2.6.5.1.	Xine [44].....	50
2.6.5.2.	MPlayer [45].....	51
2.6.6.	Writing tools.....	52
2.6.6.1.	ABIWord [23].....	52
2.6.6.2.	Open Office [47].....	52
2.6.6.3.	Kate (KDE Advanced Text Editor) [46].....	53
3.	Chapter Three – Feasibility study.....	54
3.1.	Introduction.....	54
3.2.	Software.....	56
3.2.1.	Windows 98.....	57
3.2.1.2.	Windows 98 installation suite.....	58
3.2.2.	Slackware Linux.....	58
3.2.2.1.	Better Comprehension of the OS.....	58
3.2.2.2.	Less overhead of system resources.....	58

3.2.2.3.	Better for low-end computers.....	58
3.2.2.4.	Unaltered packages.....	59
3.2.2.5.	Slackware 10.0 installation suite.....	59
3.3.	Hardware.....	60
3.4.	NC546.....	61
3.4.2.	Windows.....	62
3.4.2.2.	Problems before additional software was installed.....	62
3.4.2.3.	Conclusion.....	65
3.4.3.	Linux.....	65
3.4.3.2.	Problems before additional software was installed.....	66
3.4.3.3.	Conclusion.....	67
3.4.4.	Conclusion of NC546.....	68
3.5.	ePanel MGX.....	69
3.5.2.	Windows.....	70
3.5.2.2.	Problems before additional software were installed.....	70
3.5.2.3.	Conclusion.....	72
3.5.3.	Linux.....	73
3.5.3.2.	Problems before additional software was installed.....	73
3.5.3.3.	Conclusion.....	76
3.5.4.	Conclusion of ePanel MGX.....	77
3.6.	NC677.....	77
3.6.2.	Windows.....	79
3.6.2.2.	Problems before additional software was installed.....	79
3.6.3.	Linux.....	81
3.6.3.2.	Problems before additional software was installed.....	81
3.6.4.	Conclusion of NC677.....	85
3.7.	Conclusion.....	86
4.	Chapter Four – Implementation.....	88
4.1.	Introduction.....	88
4.1.1.	Fixing the CPU-usage problem.....	88
4.1.2.	First remarks on optimization.....	90
4.1.3.	Choosing a distribution.....	90
4.1.3.1.	Reliability.....	91
4.1.3.2.	Efficiency.....	91
4.1.3.3.	Maintainability.....	91
4.1.3.4.	Size.....	92
4.2.	Gentoo.....	92
4.2.1.	What features are available.....	93
4.2.2.	Basic Gentoo installation process.....	93
4.2.2.1.	Stage one.....	94
4.2.2.2.	Stage two.....	94
4.2.2.3.	Stage three.....	94
4.2.3.	Implementation of Gentoo.....	94
4.2.3.1.	Partition the disk.....	94
4.2.3.2.	Extract stage one tarball.....	95
4.2.3.3.	Building the Kernel.....	95
4.2.3.4.	Installing system services.....	95
4.2.3.5.	Installing all other software.....	96
4.2.4.	Gentoo – Concluding remarks.....	96

4.3. LFS (Linux From Scratch).....	96
4.3.1. LFS installation process.....	97
4.3.1.1. Build the toolchain.....	97
4.3.1.2. Use the toolchain to reproduce itself.....	98
4.3.1.3. Build the rest of the base system.....	98
4.3.1.4. Build the applications needed for this project.....	99
4.3.2. LFS - Concluding remarks.....	99
4.4. Mini-Slack.....	100
4.4.1. Installation of Mini-Slack.....	100
4.4.1.2. Compile the kernel.....	103
4.4.1.3. Compiling the Savage4 drivers.....	103
4.4.1.4. Compiling user applications from source.....	103
4.4.2. Mini-Slack - Concluding remarks.....	104
4.5. Conclusion.....	105
5. Chapter Five - Evaluation.....	107
5.1. Introduction.....	107
5.2. How good is the design?.....	107
5.2.1. Hardware.....	107
5.2.1.1. Packaging.....	107
5.2.1.2. Heat.....	108
5.2.1.3. Power Consumption.....	108
5.2.1.4. Battery life.....	109
5.2.1.5. Weight.....	109
5.2.1.6. Hardware Plug-ins.....	109
5.2.1.7. Price.....	110
5.2.2. Software.....	111
5.3. How good does this system perform?.....	111
5.4. How can this system be used in its final environment?.....	113
5.4.1. CDROM.....	113
5.4.2. Mouse: "Core pointer device".....	114
5.4.3. Keyboard: "Core character input device".....	114
5.4.4. Storage media.....	114
5.4.5. Processor.....	114
5.4.6. Screen.....	115
5.5. People's views of this system.....	115
5.5.1. People with a strong computer background.....	116
5.5.2. People without a computer background.....	116
5.6. Conclusion.....	116
6. Chapter Six - Conclusion.....	117
6.1. Introduction.....	117
6.2. Comparison with recent products.....	118
6.2.1. Comparison with the MAC-mini.....	118
6.2.2. Comparison with the Wintel-mini.....	119
6.2.3. Comparison with the \$100 PC.....	119
6.3. Conclusion.....	120
6.3.1. Power supplies.....	121
6.3.2. Price considerations.....	121
6.3.3. Which is better: Windows or Linux?.....	122
7. Bibliography.....	123

8. Appendix A – Compiling the kernel.....	127
9. Appendix B – Installing the Savage4 drivers.....	128
10. Appendix C – Compiling programs on Mini-Slack.....	130
10.1. XFCE.....	130
10.2. Mozilla-Firefox.....	130
10.2.1. Configure switches.....	131
10.3. ABI-Word.....	131
10.4. BMP.....	131
11. Appendix D – Data DVD.....	132

List of Tables and Figures

Figure 1: Soft systems thinking simplification.....	7
Table 1: Soft systems thinking applied to this project.....	8
Figure 2: The whole system.....	11
Table 2 : Prices of some available portable computing systems.....	21
Table 3: Table of eBook readers.....	22
Table 4: Summary of off-the-shelf products.....	24
Table 5: Important features of a processor board and their uses.....	27
Table 6: Comparison of mass storage media.....	28
Table 7: LCD vs. CRT.....	29
Table 8: OS comparison against common features.....	39
Table 9: OS comparison against desirable features.....	40
Table 10: Minimum system requirements for Windows [48].....	57
Table 11: On board features of the NC546.....	61
Table 12: Windows on NC546 before drivers.....	62
Table 13: Windows on NC546 after drivers.....	65
Table 14: Linux on NC546 before drivers.....	66
Table 15: Linux on NC546 after drivers.....	67
Table 16: On board features of the ePanel MGX.....	69
Table 17: Windows on ePanel MGX before drivers.....	70
Table 18: Windows on ePanel MGX after drivers.....	72
Table 19: Linux on ePanel MGX before drivers.....	73
Table 20: Linux on ePanel MGX after drivers.....	76
Table 21: On board features of the NC677.....	78
Table 22: Windows on NC677 before drivers.....	79
Table 23: Windows on the NC677 after drivers.....	80
Table 24: Linux on NC677 before drivers.....	81
Table 25: Linux NC677 after drivers.....	85
Table 26: Linux vs. Windows with respect to hardware support.....	87
Table 27: GNU Optimization variables for the VIA C3 processor.....	89
Table 28: Gentoo file systems.....	95
Table 29: Linux from scratch core packages.....	98
Table 30: Our base Linux system.....	102
Table 31: Comparison of the optimized distributions.....	105
Table 32: The product's total power consumption.....	108
Table 33: The product's estimated total weight.....	109
Table 34: The product's estimated total cost of quantities of 1000 units.....	110
Table 35: Benchmarking.....	112
Table 36: Comparison against our specifications.....	119
Table 37: Comparison against our specifications.....	120
Table 38: Which is better: Windows or Linux?.....	122

1. Chapter One – Problem Statement

1.1. Digital Divide

The Digital Divide is a trend that has been recognized for decades and have received attention in recent years [9]. The Digital Divide may be seen as the gap between those people who have access to the new Information and Communications Technologies (ICT's) and those who do not [4], [5], [9]. Community is linked with a notion of communication. Thus, one has a certain identity when one can communicate, and even more if one can communicate through the Internet [5]. Internet is a communication system that looks beyond the geographical placement of any number of entities and enables a fast and effective communication link. The Digital Divide does not only relate to the people who are not digitally connected to the Internet but is extended to encompass all information and communication technologies (ICT) [4]. This scenario exists because the technology is developed for people who are able to pay for it [3]. Developing countries do not possess the computer literacy or the money to pay for that literacy, and because technology advances rapidly the illiteracy grows daily. Furthermore developing countries are left in the dark when it comes to advances in technology simply because they import knowledge and don't generate it by themselves [5].

Existing projects showed that the Digital Divide is not only dependent on technology itself, but relies heavily on what the social context of use for the different technologies are. There exists a noticeable Digital Divide that continues to persist between people with different demographic characteristics such as age, income, education, ethnicity, and marital status [4].

Communicating over a distance is difficult if not impossible without technology. The scenario is only worsened by some peoples' bias to certain technologies or their inability to obtain technology – as described by the Digital Divide. For example: Some people may complain that they know nothing about a certain technology, but when they get the chance to learn about that technology some of them are reluctant and/or negative about learning. Hopefully this will not be the case for students who pay for the opportunity to learn.

The question now arises:

How do we educate people over a distance and across the Digital Divide?

One major problem with the set up of distance learning through Internet and ISDN may very well be the high cost and low availability. People in rural areas have little or no access to phone lines not to mention Internet or ISDN. Rural areas rarely have the infrastructure for these types of communication systems. This is due to the fact that people in rural areas are generally “poor¹” with the effect that they are not able to pay for telecommunication services. Subsidized projects that aim to bring the Internet to rural areas do exist and people do benefit greatly from these projects [3].

1.2. Distance learning

Distance learning is a new trend in education [6]. In recent years there has been an explosion of educational institutions offering their courses and even entire degrees asynchronously² via the Internet [7]. Internet is accelerating distance learning rapidly. The Internet together with multimedia³ technologies provides a paradise of information exchange [6]. When introduced to web-based learning, students are faced with technology and/or new implementations of technology [7]. CSCL (Computer-Supported Collaborative Learning) environments are often promoted as an open, safe and trustworthy learning

1 Being poor means that one not only possess limited recourses, but also little or no knowledge. [5]

2 Asynchronous means that people do not necessarily have to work at the same time or even at the same pace.

3 Media that appeals to more than one sense. Most common is seeing and hearing.

environment that produces equal opportunities for learners to participate without the limitations of knowledge levels. CSCL is recommended for distance education [8].

The main advantages of distance learning is that the student(s) and lecturer(s) are separated in time and space (temporally and spatially)[6] meaning there exists no fixed time and/or place for students and lecturers to meet [7]. Being separated by time allows any student to rerun any lecture at any time he or she chooses. This improvisation caters for slow students that find it hard to learn the first time round and for part time students which cannot attend class on a regular basis or in office hours. Being separated in time also makes it easy to rerun any missed lecture which is not possible in the traditional classroom scenario. On the other hand being separated by space allows disabled persons and people at a remote location to study along at this "Virtual University" [6]. It is also possible to monitor each students' progress and problem areas individually. Appropriate guidance can then be supplied automatically or semi-automatically. To use these advantages, sophisticated computer software is needed [6].

Despite these enormous advantages there persists some inalienable truths about distance learning: As described in the previous section, Internet is not readily available to rural areas, this is however exactly one of the places in dire need of education. The problems lie in insufficient computer network infrastructure and underlying software [6].

There exists some other drawbacks with respect to the presentation of the lectures. In the traditional university students can be grouped together to perform certain tasks and to learn from each other, this is generally not possible in the virtual university because of the fact that students are not only separated in time and space from the university but also separated from each other. Internet chat may be used for students to work together provided that they both have Internet access at the same time. In the traditional university lecturers provide on-site supervision as one of many provided functions, in

virtual universities lecturers can only provide on-site supervision and none of the other available functions [6].

There are universities that merely distribute the study material over a network together with a set of multiple-choice questions [1]. This specific scenario suggests the need for computer based learning.

The use of computer based learning proved exceptionally effective [8]. The fact that one person can rerun the study material endlessly and at any time he/she wants creates a new and unique learning environment. This also creates a different environment for each student and also stimulates the learning through repetition, which is seldom - if ever - possible in the time given at lectures.

In distance learning it may be possible to exploit the advantages mentioned above when using a multimedia device. Multimedia gives you an experience seldom seen in classrooms. For example, how many times does a student see moving pictures or hear sounds of what he/she is learning about? Multimedia gives the student this added advantage. Sometimes just seeing a picture of what one is learning about helps. Learning with multimedia is much more fun and it brings the classroom to the students' own home [2].

This is where this project comes into being. Bringing the classroom to rural areas is possible. What if we can bring the classroom to every home, and especially to everyone in the rural community?

1.3. Problem history

At the North West University they are experimenting with using a VCD (Video Compact Disk) to distribute study material for distance learning. The reason for the VCD is because there exists no form of computerized network to distribute study material. This VCD is then played on any DVD player (Digital Versatile Disk player). We want the process to be more computer based, meaning we want a bit more complexity than simple "play the video" functionality.

With distance learning, the current distribution of study material is mainly through textbooks and printed study guides. The VCD-project mentioned above has the advantage of saving space and paper. This is possible because many text books and study guides can fit on one single CD. One drawback of the VCD-project is that notes made in accordance to the study material must be on a separate piece of paper or a separate medium. Using a VCD only provides for those people that own a television together with a DVD player or have access to both. We want to produce a computer based product that enables students to give more effort to their study material than technical problems surrounding the material, while still achieving advantages associated with computer based learning.

The success of this project may reside not only in the low cost technology nor in the low cost delivery system but rather in a combination of the two [3]. This device has to be easy to use by people with little or no technological background. If it is for example possible to send a huge chunk of information on an inexpensive medium, to a low technology device, we have a good start to solving the problem. This project (starting with distance learning) may be one of the first small steps in the attempt to narrow the digital divide in distance learning.

The multimedia reader, which will eventually become the next generation of the VCD-project, will support the study material currently used by the VCD method and more.

Eventual features should include:

1. Use a CDROM disk as input for all media to be shown.
2. Display Text⁴;
3. Display Pictures;
4. Display Graphs;
5. Display Video⁴;
6. Have sound output and
7. Capability for making notes on this device.
8. Some usability beyond the classroom⁵.

1.4. Problem extraction

One problem in the traditional scientific approach is that the user does not always know what he/she wants. We propose the Soft Systems Approach to address this problem. The soft system approach describes the user's perception of the real world. Although the user's perception may differ from other people, the differences do not indicate an unsolvable question. The difference rather helps to better understand the problem situation.

1.4.1. Soft systems thinking

In systems thinking as such, the problem is seen as a system with numerous components or subsystems. These components or subsystems must all work in harmony to achieve a mutual goal. All components that contribute to this mutual goal are part of the system. Components that are not part of the system but influence the state of the system are part of the environment. Thus, a systems' environment consists of all variables that can affect its state. External components that do not affect a systems' state are not part of its environment. The environment in effect determines how the system performs.

4 Feature already usable with VCD method.

5 System must be customizable to the users' content.

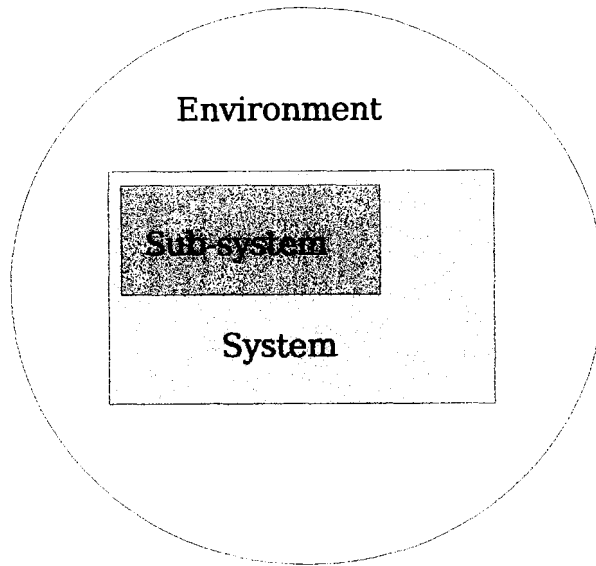


Figure 1: Soft systems thinking simplification

Figure 1 shows the system that is built up from one or many sub-systems. The system exists within an environment. The environment does not effect the (internal) working of the system directly but it can change the state of the system.

The decision maker needs to be identified before anyone can describe the problem as a system. The decision maker will be able to describe the objective of the system, which is difficult to determine. The decision maker in this case will be the project leader. In this instance the problem is to show information from some obscure medium that is not readable without the use of technology. However we need to keep the technology level as low as possible.

There exists three parts to any system in soft systems thinking:

1. System
2. Subsystems or components
3. Environment

These three components can be seen in figure 1.

Table 1: Soft systems thinking applied to this project.

Subsystems or components:	Environment:
1. Software	1. Users
2. Hardware	2. CDROM

Table 1 shows a very simple assessment of the system. Firstly there are the Subsystems and the Environment. Each of which is divided into their own subsystems.

All parts of the system must work together in harmony to achieve the mutual goal. The goal is in short: "To present enriched⁶ study material to the student." The system has the responsibility to read CDROMs together with some other optional medium that will be used to store additional information and display the two together in an orderly fashion.

Due to the human nature of wanting more than we already have, keeping technology low may prove very difficult. New technologies are developed and/or are marketed on an almost daily basis. What is an up-to-date system today is ancient next week. Designing a system with low technology and enabling it to "evolve" along with other technology may just be a start to bridging the Digital Divide. This may enable some users, who are on the "wrong" side of the divide, to teach themselves more about technology while it "evolves".

It may be naive to think that it is possible for all people to cross the digital divide, although it might just be possible to narrow the divide.

1.4.1.2. The Subsystems or components

Table 1 indicates that the subsystems or components are the hardware and software. The software is not a component, but rather a whole subsystem, which may more effectively be referred to as the "operating system". The other

⁶ Enriched in the sense of multimedia. Study material that is able to draw more attention to itself.

software that we will use are all subsystems within the operating system. The operating system incorporates much functionality by itself, but more precisely it generates an environment for user application software to work in. The user will interact more closely with the user applications than the operating system.

Software subsystems include:

1. Operating system
 - 1.1. Window manager
 - 1.2. Web browser software
 - 1.3. Software to make notes
 - 1.4. Sound codecs⁷ and players
 - 1.5. Video codecs⁸ and players

The hardware are also many subsystems rather than a single component. These subsystems communicate with each other to create one single subsystem that also communicates with the software subsystem.

Hardware subsystems include:

1. Keyboard
2. Mouse
3. Processing unit⁹
4. Screen
5. CDROM reader
6. Screen Output

The people using this device will need easy-to-use input devices. Hardware that is both widely used and easy to use will be preferable in this case. Keyboards are used with every computer and is self explanatory. The keyboard will be one of the subsystems interacting with the environment to store notes made in accordance with the study material. The mouse, which is used with nearly every computer, exists in one form or another. The mouse is a popular tool and

7 Sound codecs to play some compressed formats such as mp3 and ogg vorbis.

8 Video codecs to play compressed formats such as DivX and Xvid.

9 The processing unit is an arithmetic unit that processes all commands given to the device.

easy to use. The mouse will be used to scroll through study material and choosing applications to be run.

1.4.1.3. The environment

The environment incorporates the users and the CDROM that will be sent to all the users. The CDROM is part of the environment because the system must make it possible to change the study material without problems.

The users will ultimately be all the pupils doing the distance-learning course(s). The pupils will change the state of the system and ultimately determine the success of the system. Meaning that if the pupils feel that they benefit from this product, the product reaches its goal.

People who do not necessarily possess the knowledge to use complex devices such as notebooks, laptops, PDAs or even personal computers will use this device. In other words, the systems' environment is not favorable. Therefore, the device needs to be user friendly and easy to use. Furthermore the system must be insensitive to incorrect inputs generated by the environment, ignoring harmful input and reacting only to useful input.

The system needs to be low cost technology and a CDROM disk will play the roll of the low cost information transfer medium. Thus, the CDROM disk will be one of the systems' variables. The CDROM disk is not part of the system but rather the environment. If The CDROM disk containing study material is changed, then the system is still the same system. Therefore, the CDROM disks are part of the environment which we expect to change as time goes by.

1.4.1.4. Summary of the systems thinking model

The system will work in an environment. It is shown in figure 2 that the environment consists of the users and the study material from the CDROM disk because both of them alter the state of the system. The system must interact with its environment therefore, it must be able to take inputs from the environment and give outputs to the environment.

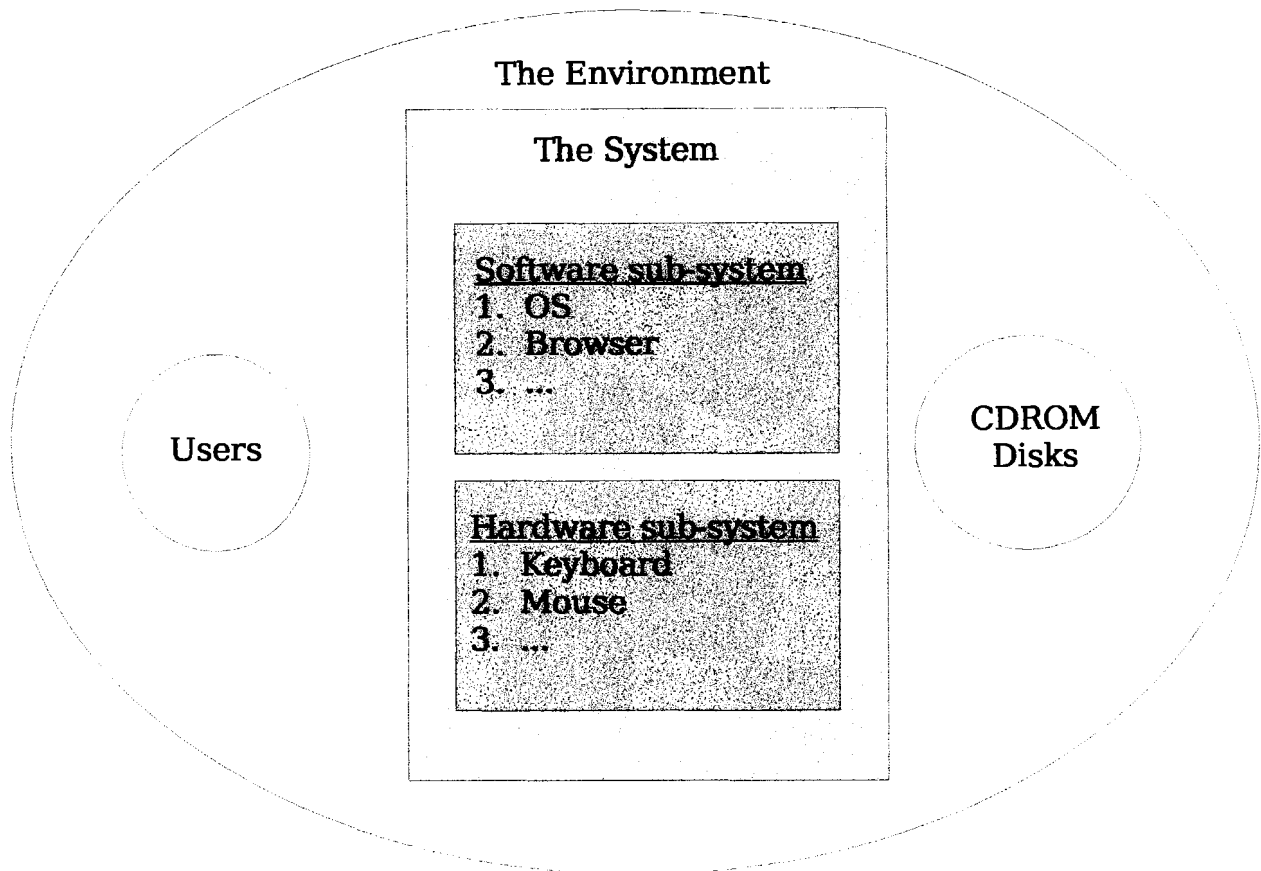


Figure 2: The whole system

The system will create its own environment for subsystems and components because it in turn alters the state of these subsystems and components. These subsystems and components consists of hardware and software. The hardware and software must communicate effectively to achieve a mutual goal. Therefore, they are components of the complete system.

1.4.2. Problem Definition

The goal of the multimedia reader is to aid a person in learning study material. We need a portable device that makes it possible to show data from a CD on a screen. The data that will be used consists of the text, still pictures, videos and sound.

1.4.3. Specifications

The formal list of specific features are as follow:

- Data read and display capability;
- Additional data storage capability (optional);
- Portable¹⁰ and lightweight¹¹;
- Sufficient screen size¹² for pictures and diagrams;
- Colour screen to enhance viewing material;
- Sound output for enhanced material;
- Text input device;
- Pointer input device;
- Sound input capability (optional);
- User friendly;
- Low cost and
- Low power usage.

The specifications given above are all necessary as discussed below. Applying systems engineering implies asking questions about the use of every component or subsystem.

1.4.3.1. Data read and display capability

Study material is sent to all distance learners via mail on a simple CDROM. This data cannot be read without technology and this device must incorporate a CD reading capability. The data is in digital format, thus we need technology to show this data on a screen. This screen must be of acceptable size and resolution.

Data to be shown

The main object of this project is to assist in the learning of study material for distance learners. Therefore, the data that is to be shown will include text, pictures and videos.

10 Portable means that it is not tied down by any requirements such as power supply.

11 Lightweight in this case emphasizes the ability to be carried around effortlessly.

12 The specifications are for at least A5 size (148mm X 210mm) screen.

This multimedia reader need not support DVD playback at very high quality picture and high resolution. Taking out such features lowers the cost considerably while still enabling acceptable video quality for the task at hand.

1.4.3.2. Data storage capability

Notes made in accordance to any study material can be made on any piece of paper or in an exercise book. This is not ideal, and storage of notes on the multimedia reader can be considered as an option.

If this function is only used for text (as most notes will be) then we will not use much space for any notes made. The question now arises: "How much space will suffice?"

Storing data also enables this device to store additional programs therefore, enabling it to grow into a complex unit enabling its users to customize and shape it into what they really want. This device is then able to "grow" with its users, while they learn to use technology.

1.4.3.3. Portable¹³ and lightweight¹⁴

Since this multimedia reader is aimed at distance learning we must cater for all conditions under which the learning can possibly take place. This is not easy. Away from home, travelling or even at home where there may or may not be electricity. It is obvious to say that some people may not need a portable learning tool. It is easier to make a portable and lightweight tool stationary than to make a huge and heavy machine portable. The included battery pack will prove extremely useful in emergencies when there is a power failure.

13 Portable means that it is not tied down by any requirements such as power supply.

14 Lightweight in this case emphasizes the ability to be carried around effortlessly.

1.4.3.4. Sufficient screen size¹⁵

It was stated earlier that a screen is necessary. We now need to consider the size of the screen. Scrolling through text is annoying if you can only view four lines at a time. It is much more practical to be able to view large amounts of text at once.

The screen size is also important for viewing full pictures and especially graphs. Scrolling to and fro over any large picture just does not give the whole idea. Seeing the whole picture at once is more desirable. Pictures that will be used in study material rarely cover more than one half of an A4 page. The specification of A5 (or 148x210mm) is stated by the client. This translates to roughly 10" screen size measured diagonally with a 4:3¹⁶ ratio.

The specification stated is not necessarily the best choice. Given technology and the effect of mass production on its prices, it may cost the same for a 10.4" and a 12.5" even though the one is larger than the other.

1.4.3.5. Colour screen to enhance viewing material

Viewing black and white pictures and graphs in textbooks may be acceptable. That may not be the case with videos. Technology is advanced enough to supply colour displays for a fairly acceptable price. For this reason monochrome displays will be out of the question.

1.4.3.6. Sound output for enhanced material

Sound is one of the underlying qualities that create the full multimedia effect. Multimedia without sound would just not be worth the effort. The device should be able to play all of the most widely used formats that is available today.

Many different sound formats exist today. This is one of the areas that advanced a few years back due to the demand for more sound with smaller

¹⁵ The specifications are for at least A5 size (148mm X 210mm) screen.

¹⁶ The ratio of 4:3 should be interpreted as 4 units horizontal by 3 units vertical.

storage space. When people started recording their personal CD music on computer, hard disk space was very limited. Therefore, advanced compression techniques were developed. Unfortunately this also gave rise to easier piracy.

Many different compression techniques exist today such as MP3, OGG Vorbis, FLAC, etc. MP3 and OGG compresses sound to a ratio of around 10 to 1 with almost no difference in sound quality to the human ear although OGG files take up less space for the same sound quality [30], [31], [32]. Most users and even artists are not aware that MP3 is not free. Thomson electronics and the Fraunhofer institute jointly own the MP3 patents and charges royalties for their use while OGG is free [28], [29], [30], [31], [32]. MP3 uses much processing power for encoding (compressing) as well as decoding (listening). OGG Vorbis on the other hand uses slightly more processing power for encoding (compression) and less for decoding (listening) because you only compress it once but listen to it more than once. FLAC (Free Lossless Audio Codec) is a lossless audio codec suitable for preserving audio without sacrificing audio quality [28]. FLAC compresses sound to roughly half of the original size.

WAVE files are not often used today because it takes up too much disk space. Although WAVE files can be in a compressed state, they are generally not compressed and consist of raw PCM (Pulse Code Modulation) with a small header written in the beginning of the file. The header gives information about the characteristics of the PCM to follow, such as length, bits per sample, mono/stereo and if ADPCM (Adaptive Differential PCM) compression is used. The header also tells the OS if the file is a WAVE file or something else. That is if the OS looks at the header of every file, and not just the extension!

1.4.3.7. Text input device

The use of storage media was discussed earlier. Storage media for storing notes made by the students would be rather useless without a text input device such as a keyboard or even an on screen keyboard.

Making notes is one of the advantages that may be exploited with this multimedia reader.

1.4.3.8. Pointer input device

The computer mouse has been around for almost twenty years and it is hard to imagine a computer without one. A pointing device has many uses and does not add much to the budget.

1.4.3.9. Sound input capability (Optional)

This function is not intended for voice recording or even voice recognition, although that would be possible within the next few years.

VoIP (or Voice Over Internet Protocol) is becoming more popular. This new development enables people to communicate by means of sound (or voice transmission) over a normal digital network which is capable of Internet Protocol, much like telephone service such as a dial up connection, but which is more cost effective. Communicating with other students when and if there is a network connection between two parties may prove useful when students must work together on some tasks given by lecturers.

Note that the original specifications does not imply an available network therefore, this only enables future upgrades.

1.4.3.10. User friendly

Being user friendly may be a feature rather than a specification. This is indeed necessary because some of the distance learners may never have worked with any form of computer in their whole life.

Being easy to use is one of the constraints therefore, it is one of the specifications.

1.4.3.11. Low cost

The cost is important, otherwise other readily available “off-the-shelf” devices such as laptops or palmtops could have been used.

This device must enable the user to perform certain tasks associated with the study material. Thus we need the basic functionality without the unneeded (and sometimes expensive) features, this is not easily obtainable with “off-the-shelf” products because these products cover a much wider range of uses, ranging from school to the professional world.

1.4.3.12. Low power usage

Conserving power is one major advantage. Enabling this device to run from battery enables distance learners in rural areas and people who travel frequently to continue working without external electricity for reasonable lengths of time.

1.5. Conclusion

This device may be used by many a person, ranging from distance learners to the lecturers themselves. That makes this project difficult.

This project not only goes one step further than the VCD project described to enable learners to make notes on this specific reader, but also adds screen resolution, expandability and portability.

What will come after this? Students may want to change the look and feel of their own personal multimedia reader. It is possible to change the background, the screen saver, the icons, the system sounds and even the window manager. It is possible to install new programs, to upgrade existing programs and to even change programs that do one function with another similar program that better speaks to your personality.

Producing a device that enables upgrades (as opposed to a total rework) to further develop its functionality is the key. Human nature dictates that we

strive for more than we already possess. Therefore, we propose a project that is able to “evolve” as the person using the device sets forth in crossing the Digital Divide. Each time the student learns a bit more about the technology at hand he can add a little piece to the technology making it better in his/her opinion. This may create a custom learning environment which may ultimately help many students to cross the Digital Divide.

1.6. Overview of the rest of this document

Chapter two will list and discuss some of the available hardware and software at the time of this writing. Comparisons will be drawn between some available hardware and also between some available software.

Chapter three will compare Windows and Linux on three readily available processor boards. The two OS's will be compared with respect to the specifications given in this chapter.

Chapter four shows how Linux was optimized to the specific task at hand. Linux was optimized for speed and overall disk space. Optimizing the speed enables us to use less expensive hardware to obtain the same results. Disk space is important because it enables us to allocate more free space for users.

Chapter five gives an assessment of the system as a whole. This chapter also compares some physical features and tries to give some helpful advice for future products. This chapter also includes some comments from various people who used this product for a brief time.

Chapter six compares the device with other newly available products at the end of this project. Chapter six also concludes on the whole project.

2. Chapter Two – Literature study

2.1. Introduction

To solve any physical problem we need hardware – the users will need a physical object to hold and interact with. Without hardware to run on, software is quite useless. This section will look at possible hardware for starting to solve the problem and some software possibilities for assisting the hardware in its task. All hardware will be measured against the requirements given in chapter one, because software cannot enable features that are not supported by the underlying hardware.

Most modern hardware need software to become functionally usable. The software determines how the hardware will solve the problem. The software and more accurately, the Operating System determines the look and feel of the virtual environment within this hardware. Some operating systems will dictate which software can be used and which software can't be used. The operating system will also determine the stability of the entire system.

2.2. Hardware (What is available)

First of all we need to consider existing products. The existing products show what the market demands. The existing products have much research in them to show what is necessary to implement and what not.

Secondly we need to consider the price and availability. Price places a large constraint on the project, because this new device will be developed for low

cost. Note, from chapter one, that this device is primarily intended for people who live in rural areas and cannot afford a state-of-the-art machine.

Thirdly, if there are no devices that just quite satisfy the project objective, we construct an entirely new device to do so. A new device may consist of parts of other products or in this case even a downscaled version of an existing product. The latter implies a computer, we need to downscale a standard desktop computer or laptop to compensate for cost. Furthermore, we do not need a fast computer for this current application, we merely need a computer that is fast enough to satisfy our need.

2.2.1. Portable computing systems

Some portable computing systems such as laptops/notebooks and PDA's are popular systems which can be used independent of other computing systems. Laptops are portable versions and variations of the desktop PC therefore, they all satisfy the requirements for this project save the price range. Throughout the course of this project laptop prices lowered considerably and some new laptops are now cheap enough to be used for this project. Second-hand laptops can be acquired well within the budget, however stock is easily depleted.

A PDA is a pocket PC which is often used in addition to the desktop computer, for example, work can be done on a PDA and be synchronized with data on a computer at a later stage. Very few of these devices make it possible to read from a mass storage device such as a CD-ROM. If they are used in addition to the desktop PC then mass storage functionality would be unneeded. Furthermore some PDA's and notebooks that do satisfy all of the requirements are beyond the budget of this project. The high price of a PDA is due to the fact that most of them come with touch screens that make these little machines pretty expensive. PDA's also do not have large enough screens to satisfy the requirements of this project.

There is too large a number of different PDA's (Personal Digital Assistants) to choose from. There is not enough space to insert every known PDA. For that reason Table 2 will only state two randomly selected PDA's for comparison of needed features.

Table 2 : Prices of some available portable computing systems¹⁷

Name	Screen	Sound	Portable	CDROM	Other I/O	Budget
Acer TravelMate 260 Laptop	14" Colour	Yes	Yes	Yes	Yes	Yes R3000
Asus A35LC14CM Laptop	15.1" Colour	Yes	Yes	Yes	Yes	No R7000
Fujitsu Siemens Pocket LOOX 410 ENG PDA	3.5" Colour	Yes	Yes	No	Yes	Yes R3000
Asus MyPal A730W PDA	3.7" Colour	Yes	Yes	No	Yes	No R4300

2.2.2. eBook readers

EBooks are coming back into the market after the "dot com" collapsed in the late 1990s. These small electronic book readers are portable and bring down the cost of books by an estimated 95 percent [10]. Imagine having to carry 100 books from one bookcase to another room in the house. Now all books are on one eBook reader and carrying them all around even on vacation is as easy as carrying your cellphone with you.

EBooks have a great advantage with storage space and help save the trees and environment by using no paper as opposed to printing all the books separately.

¹⁷ Prices vary easily and are a snapshot of 30 August 2005 [13].

Many programs exist that gives the functionality of a eBook reader for PC's, Palmtops, PDA's and Cellphones. These software include Plucker, iSilo, Weasel Reader, eBook for Nokia [11].

Table 3: Table of eBook readers.

Name	Screen	Sound	Portable	CDROM	Other I/O	Budget
GEB 1150	5.5"	No	Yes	No	Modem	Discontinued
GEB 2150	8.2"	No	Yes	No	Modem	Discontinued

Ebook readers had full colour screens of 5.5" up to 8.2" [12]. For this project, eBooks will not be suitable because it only supports text. Media must be downloaded from the Internet instead of a mass storage device such as CDROM. Neither Ebooks nor eBook readers support multimedia. Also Gemstar closed their ebook reader production in 2003 [12], thus using discontinued products would cause problems in the near future.

2.2.3. Portable DVD

DVD players have ample screen size, although they lack the possibility to be programmed and the screen's quality is not good enough for reading high volumes of text as would be the case with study material. This portable entertainment system might be good enough when setting forth the VCD project with notes made separately, however, we want something more general than the current VCD-player.

This device lacks the possibility to store additional notes made by each individual student to his/her own study material therefore, we disregard this device from the current options.

2.2.4. The Desktop PC

Personal computers and laptops are very customizable. Anyone can get a customized quote for a personal computer from R3000 and up, depending on what you want. Peripherals can be added with add-on cards. Desktop computers have the ability to add a lot of extra hardware on its many open slots. These add-on cards include network cards, modems, sound cards, display cards, data acquisition card and much more. The ability to add hardware easily, expands the usefulness of desktop computers almost endlessly.

Personal computers have all the functionality needed and more. The problem, in today's computers, is that there exists too much embedded functionality such as on board network and hardware raid controllers, this fact influences the cost. Some built-in features are necessary and some never get used. Second-hand desktop computers are within the budget, however, this stock is easily depleted. Desktop computers are also not portable!

Laptops were beyond the budget at the start of this research, even though they are portable. Laptop prices came down by as much as 50% in the course of this project. Laptops also have a wide range of customizable features. Depending on the features, new laptops currently sell at around R3000 and up.

Personal computers have input options such as a keyboard and the ever-popular mouse. Personal computers also include a CDROM option and a data storage options almost as standard. These computers have more than big enough screens.

The drawbacks of these computers are that they are not portable, even though they might fit comfortably into the budget. Laptops are just beyond the budget, however, they are a bit expensive and a bit too large. Any standard laptop or desktop computer will more than satisfy our needs, however the cost, size and functionality is too high.

2.2.5. Summary

The summary of commercially available products tries to put the information given thus far into perspective. Products that do fit into the budget do not satisfy the need.

Table 4 shows a rough comparison between the different hardware discussed thus far.

Table 4: Summary of off-the-shelf products

Product	Specifications	Portable	Cost (In budget)
Cheaper PDAs	No	Yes	Yes
More expensive PDAs	Yes	Yes	No
Ebook readers	No	Yes	No (Discontinued)
Portable DVD players	No	Yes	Yes
Desktop Computers	Yes	No	Yes
Laptops	Yes	Yes	Yes

2.2.6. New development

This project is indeed in need of some of the functionalities stated in all of the readily available devices. However, this project needs somewhat less functionality of each available device. Each device has a quality that attracts interest to it.

Storage is available in all the devices, but not the same storage. We do not need the mass storage available on hard disk, less than one Gb of data will suffice. We need screen output, which is also available in all devices. Input is not the same for all devices. We need an easy way of text input and input for control. We also want the device to be low-cost. Therefore, we basically need a

specialized machine. This machine will do only one task and students must only pay for that one task and no other unneeded features.

A new development may be quite foolish when considering the human nature. Ask yourself: "How long is my own computer good enough?" Each time you get a new tool it is good enough for a while and then you want more. Keeping this in mind, the new development needs to be at least software/firmware upgradeable. One way of ensuring that your hardware will stay capable is using the latest hardware and software developments. For example, simply using USB devices rather than serial and parallel ports is an option that does not necessarily mean more money.

Furthermore, building any custom device has an initial high development cost and requires large amounts of time. Developing a custom device from scratch will be too expensive. Therefore, this project will focus on the integration of standard off-the-shelf subsystems¹⁸ with each other. Standard off-the-shelf components ensure high quality hardware that is extensively tested and stable.

All of the off-the-shelf devices have some common features. All of them allow some form of mathematical and logical processing. Most of them have storage capabilities. All incorporate some form of input and output (or I/O for short). Whether we know it or not, all of them require software to enable the functions of the following subsystems:

- Processing unit;
- Mass storage media;
- Inputs and Outputs and
- Software¹⁹

¹⁸ Subsystems include hardware interfaced with other hardware and also hardware interfaced with software.

¹⁹ Software does enable software. This will be better explained in the section about Operating Systems.

2.2.6.1. Comparison with a general purpose desktop PC

It may be a good idea to discuss the new development as opposed to readily available hardware such as the desktop PC. This new development will consist of some features available to normal PC's.

This new development must be portable, meaning that it can easily be carried around, whereas desktop PC's requires a considerable amount of effort to move from one room to the next. The next difference is the power consumption. This device needs to use as little power as possible if it is to accompany students on the road. Whereas power consumption was never a major issue in the desktop computer. The power consumption also relates closely to heat generation. If a desktop PC runs to hot, we have water and oil cooled systems. This is not practical for a portable system.

The software used on general purpose computers will differ from the software needed for this multimedia reader. Firstly general purpose computers must make it easy to install new software. For the multimedia reader this is not true because we know in advance what software most students will need. Easy installation of new software also makes it easy for unwanted software (such as viruses) to be installed without the user's consent. Secondly general purpose computers have stronger or faster processors than used in this project. Thus the need for optimization does not arise for most PC users. This device will have to use optimized software to assist the hardware in its task. Lastly PC's do not have "limited storage space" with respect to the OS alone. The multimedia reader will have limited storage space, this will later be overcome by selecting modular software to run on the device.

2.2.6.2. Processor board

Finding a processor board that is capable of the specifications is not as easy as it may seem. These processor boards also come with a whole list of unneeded features.

Because these processor boards are highly specialized, they do not normally enable add-on cards as normal computers do. Therefore, all of the needed hardware functionality needs to be built-in.

The processor board should incorporate a CPU and some I/O. The special features communicate through the I/O channels with the CPU. Features that are important to this project are given in Table 5.

Table 5: Important features of a processor board and their uses

Features (Type)	Peripheral	Use
PS/2 (I/O)	Keyboard / Mouse	Inputs to the reader
USB (I/O)	Various peripherals such as speakers, USB storage devices etcetera	Whatever the user wants to add by himself
IDE (I/O)	CDROM and Compact Flash interfaces	CDROM and mass storage media
Video output (I/O)	Video card	Adds the ability to connect a screen to the reader
Sound support (I/O)	Sound Card	Sound input and output
Processing unit (Processor)	CPU	Handling all processing functions necessary for normal operation

2.2.6.3. Mass storage media

Mass storage media will be used in two ways. One way is for the distribution of the study material because it cannot be distributed over a network. This study material does not have to be writeable, however it needs to be extremely low cost. The second storage media is to store user files and the operating system. This media needs to be writeable, otherwise users cannot store their own files on the media.

Table 6 shows the different mass storage media available today and compares some common features.

Table 6: Comparison of mass storage media

Storage media / capacity	Price relative to storage space	Advantages	Disadvantages
Hard disk / 40Gb and up	Low	Fast Long lifetime Proven technology	High power consumption Large physical size
CDROM / 650Mb & 700Mb	Very Low	Fast Long lifetime Hot pluggable	Read-only
Compact Flash / 64Mb up to 8Gb	High	Low power Small physical size Light weight	Slower than Hard disk Limited write cycles
USB thumb drive (Flash drive) / 32Mb up to 2Gb	High	Low power Small physical size Hot pluggable	Slower than Hard disk Limited write cycles USB connected
Disk on chip / 388Mb up to 1Gb	Extremely expensive	Fast Small physical size Low power	New technology Special support needed

Compact flash does not seem like an obvious solution in this project because it is small and expensive. One single gigabyte of compact flash costs roughly the same as 40 gigabyte of hard disk space. Compact flash on the other hand has amazingly small physical size, weight and low power consumption.

2.2.6.4. I/O devices

The inputs and outputs can be seen in Table 5. Inputs and outputs are necessary for the device to communicate with the human-world. Special I/O devices such as USB make it possible for users to connect their own peripherals

to the device creating the opportunity for users to use this device outside the classroom as well.

Stated below are the advantages and disadvantages measured against each other of LCD's and CRT's:

Table 7: LCD vs. CRT

Features	LCD	CRT
Light Weight / Portable	Yes	No
Cost	R 1500 and up (10.4")	R 800 and up (17")
Resolution	800x600	1024x768
Power usage	5 Watt	75 Watt
Power supply	5/12 V _{DC}	240 V _{AC}

In Table 7 it can be seen that although the LCD unit may not be the cheapest choice, it is portable. Portability is important to this project. We want a device that can be operated from a battery and can be carried around as the need arises. The resolution of 800x600 also satisfies the requirements.

All I/O functions must be supported by the operating system. Therefore, it is necessary to chose the operating system well.

2.3. Software

Now that the hardware has been described, software can be chosen to effectively run this device. Keeping the price low even further, asks that standard off the shelf software be used. Developing custom software has enormous initial costs in terms of paying software developers.

This device needs three main packages:

- HTML-web browser, to view the study material in an orderly fashion.
- Text editing software to write notes about the study material.
- Multimedia player to play some of the multimedia in the study material.

First of all these packages cannot run on the raw machine by default. Off-the-shelf software are not created to be run without an Operating system.

This device needs an Operating System (OS), because the user will need to be able to manipulate the hardware effectively. The full function of an OS is hard to define and shall be described in more detail later on.

2.3.1. Choosing an OS

Windows is the most popular choice covering near 90% of all desktops [18], however we want free software in order to reduce the total cost of the system. Some software offers graphical interaction with users and some do not. Some packages offer multitasking and some do not. Some offer user access rights and discrete privileges and some don't. Some Operating Systems (or rather their creators/companies) even offer good apparent support. "Some OS's are stable and need not be restarted every time someone sneezes" [18].

We have the following software specifications:

- Graphical User Interface;
- 24/7 running time;
- Multimedia Capabilities;
- Time sharing between processes;
- Storage capabilities.

2.3.1.1. Graphical User Interface

The graphical interface has proven to be very popular in the computer world. This can be due to the fact that very little people know and understand computer commands in the console/shell (text based interaction).

The graphical user interface allows people to use the core pointer (such as the mouse) more extensively. This enables the user to click on an icon and some task is automatically performed by the software.

2.3.1.2. 24/7 running time

Some students have the ability to work for extended periods of time on their study material. Therefore, this device needs to be able to keep up when necessary. Some OS's cannot keep running for extended periods of time and needs to be restarted regularly.

2.3.1.3. Multimedia Capabilities

The hardware is chosen to support multimedia. Now the OS needs to support this function as well. Today very few OS's lack multimedia support therefore, this is not a very important option to keep in mind.

2.3.1.4. Time sharing between processes

This device needs to be able to support multiple programs at once. Reading ones' study material and then closing that application to write the notes are completely unacceptable. Almost every OS to date supports time sharing between processes.

2.3.1.5. Storage capabilities

This option allows disk access to the user's applications which will enable the user to write notes and also to review those notes at a later stage.

The storage capabilities also enables applications to be stored on the disk. This enables users to install their own applications at random. Therefore, allowing users to customize their multimedia reader to their hearts content.

2.3.2. What is an OS

An operating system (OS) is an important part of almost every computing system. This is because the OS transforms the hardware into a favorable environment for other applications to exist.

A computer can be divided into four components: the hardware, the operating system, application programs and users [16].

An OS has to be able to run all types of programs ranging from simple calculations to word processing to games and even virtual reality. Therefore, it has to be greatly generalized.

An OS has two design objectives:

1. Virtual machine

The raw machine, or personal computer (PC) is not usable to the standard user. For this reason, we need an OS to transform this hardware into a virtual "World", whose characteristics are somewhat different from the physical machine [14].

2. Value for money

This means that the OS must make the best possible use of the physical hardware [14].

2.3.3. The function of an OS

It would seem a fairly simple task to define what a certain piece of software is supposed to do and how it should be done. In the case of an OS the situation is complicated by different people's perception [15]. For example, some people think the OS lets you surf the WEB – which is not true. To understand operating systems we have to reassess what the computers' evolution has gone through the past 30 years. The short answer is that: "An OS provides an environment within which other applications can do meaningful work" [16]. In other words the OS creates a working environment which is not hardware dependent and is more usable than the raw hardware on which it runs.

The OS has hidden processes that are critical to the working of modern computing systems such as [14], [15], [16]:

- Manage the processor to service several running processes in parallel;
- Memory allocation and management by allocating resources to programs when they need it;
- Running other programs, executing and terminating programs;

- I/O (Input/Output) handling and
- Data access such as file and network access.

Functions that are usually considered part of the OS include [15]:

1. Functions to support device mounting, system configuration and reconfiguration, and system status modification. This interface allows a user to execute and kill applications, inquire about hardware utilization, inquire about the rate of progress of a particular program, inform the system of the association or disassociation of data sets with various devices, etc.;
2. Handling the basic I/O activity of the system in question. Programs that manage queues of I/O requests, submit I/O requests to devices, handle interrupts from I/O devices, and allocate devices and channels for the use of a particular programs or the residence of a particular data sets;
3. The functions that allocates processor time to currently running programs. The allocation of services and rates of service is frequently thought part of the OS, however this may be thought of as an extension of the complete system.
4. Those mechanisms that provide the methods of access to computational resource. For example, control streams, readers/interpreters and interactive terminal command interpreters (Such as the “born again shell” or BASH).

The greatest challenge to OS designers is to strike a balance between generality and efficiency [15]. In terms of OS requirements, the general publics' needs are different from the needs of this specific project. Therefore, we must ask the question: “Is it possible to create our own custom OS for our own custom problem?”

An OS does not let you surf the web or listen music. Whether a software application is part of the OS or not is partly a matter of judgement, partly a function of the views of the system that are exposed, and partly a function of the history of the system [15].

2.3.4. Why we need an OS

Users are primarily interested in the information they can get from the system [15]. There is no reason for everyone that uses a computer to know how the system works in all of its details.

Without an OS, all users would have to possess an immense technological knowledge of how a computer works. Even with an OS some knowledge is needed, the minimum amount of needed knowledge depends on the specific OS in question.

If a I/O intensive program and a processor intensive program can run simultaneously, the total cost of executing both can be made lower than the sum of costs of executing both programs in succession. This manner of use is widely known as time sharing [21]. Time sharing between jobs (or processes) without an OS is difficult, if not impossible, never mind running processes simultaneously and in real time [16]. Time sharing reduces the cost of computers even further – a computer that runs programs a certain part of the day and another program for the rest costs less than two separate computers [21].

The function of software (and ultimately the OS) is to maximize the economic value of computing machinery to users by [15]:

- reducing the cost to use the machinery
- assuring timely delivery of information
- increasing the effective use of machinery when the cost of hardware is non-trivial

Software fills the gap between hardware and user interfaces. Certain interfaces provide certain functions that aims to reduce the user's workload by performing tasks that otherwise would have to be done by hand.

The interface is economically sound if its total cost is less than the cost of not having the interface available. Where the total cost of the interface equals the

price of the interface plus the hardware resources it consumes. Software is economical if the cost of its use is, in fact, less than the cost of not using it [15].

2.3.5. Desirable features of an OS

2.3.5.1. Efficiency

Efficiency, for a computing system, is difficult to define on one single criteria. Sometimes an OS is measured against more than one criteria for efficiency, various possible criteria are listed below [14].

1. Mean time between jobs
2. Unused CPU time
3. Turn around time (for batch jobs)
4. Response time (in multi access systems)
5. Resource utilization

2.3.5.2. Reliability

Ideally an OS must be error free, this can however not be achieved. For a system to be reliable - not error free - it has to constantly produce the same output if the input program and data stays constant [14].

2.3.5.3. Maintainability

It should be possible to maintain, correct errors and enhance the OS without needing an army of programmers (in comparison to the system size) [14].

2.3.5.4. Small Size

The OS must be in proportion. This is the relationship between space used and functionality. No need in keeping a relatively large OS if the users need a normal typewriter. Large systems are more prone to error and takes longer time to write than smaller systems [14].

2.4. Summary of an OS

How complex is this piece of software we call an Operating System? Everyone who knows computers, has spent countless hours interacting with the OS with or without even realizing that it exists. Though everyone working with computers uses the OS constantly, few realize its complexity and importance.

Deciding where to draw the line between OS and user applications is a very difficult task. One group of persons would like permissions or access rights to be done by the OS, when other parties would like a separate application to handle access rights. System logging can be done by the OS, however, then the users cannot chose which system logger they want, thus some users would like it to be done by a custom and stand alone application.

Since the primary goal of the OS is convenience for the user, our system cannot be distributed without one.

2.4.1. The OS applied to our project

We want our OS to be enable our hardware to run graphical applications. Our OS must be able to handle multiple processes at once, such as displaying study material while enabling the user to save notes to disk at the same time.

Our OS must support optimization. We want to use low-cost hardware and therefore, the OS must be able to make effective use of the hardware available.

2.5. Different OS's

There is not enough space to write about all operating systems. This discussion will only take a few operating systems commonly known to personal computer users into account.

2.5.1.1. MS-DOS

MS-DOS (Microsoft Disk Operating System) is a program that coordinates activities on a computer, and controls every part of your computing system [20].

DOS provides two fundamental functionalities [20]:

- A method of communication with the user handling Inputs from and Outputs to the user, and executing the user's commands and
- A method of program and data storage and retrieval.

MS-DOS (Microsoft Disk Operating System) is the most common OS for 16-bit processors [16]. Everyone should agree that a lot has changed, in computers specifically, since the MS-DOS era. 32-bit Processors are relatively cheap and graphical interfaces have become a standard feature on any computational system, even Personal Digital Assistants (or PDA's) exhibit graphical capabilities.

MS-DOS is rarely seen today, however it is still not a free OS.

2.5.1.2. Microsoft Windows

Microsoft Windows (as we know it today) runs mainly on 32-bit processors. Windows is the most popular OS for standard desktop PC's, currently running on 90% of all desktops PC's. Windows is, however, not very popular for servers [18].

Viruses are a real threat to computers today, ranging from harmless access violation to serious denial of service attacks (or DOS-attacks). Viruses and worms are mostly targeted at Microsoft software [18].

Microsoft Windows is not free. Several different versions of windows range from R900 (Windows XP home) up to R6000 (Windows 2003 server).

2.5.1.3. Linux

Linux started from a man named Linus Torvalds, who started writing his own Unix-like kernel to use on his own home PC. Others encouraged (and contributed to) Linus into taking a more portable approach [18].

According to GPL and the Open Source Licensing agreement, Linux is free.

GNU (recursively named by the phrase GNU is Not UNIX) started in 1984 as a project of the free software foundation. GNU was intended to be a recording of the entire UNIX operating system that could be freely distributed [18].

Once Linus and his friends had a working Linux kernel, pulling together a complete open source operating system was possible. The reason this could be done was because so much of the available “free” software was [18]:

- Covered by the GNU Public license (GPL) or a similar license. That allowed the entire OS to be freely distributed, provided that some guidelines were followed relating to how the source code for that software was made available going forward.
- Based on UNIX-like systems. Clones of virtually all other user-level components of a UNIX system had been created. Those and other utilities and applications were built to run on UNIX or other UNIX-like systems.

Today Linux can be described as an open source UNIX-like OS that reflects a combination of SVID, POSIX and BSD [18].

2.5.1.4. UNIX

Because Linux is based on UNIX and Linux is free, UNIX will not be considered as an option. One goal of the OS is to lower the total cost of the system while still achieving full functionality.

2.5.1.5. Comparison of OS's

Table 8: OS comparison against common features

	Graphical User Interface	Uptime	Multi-media	Time Sharing	Storage	Price
Dos	Basic	Short	Basic	No	Yes	Low
Windows	Yes	Short ²⁰ to Permanent	Yes	Yes	Yes	High (±R1300)
Linux	Yes	Permanent	Yes	Yes	Yes	Free
Free BSD	Yes	Permanent	Yes	Yes	Yes	Free

Linux and BSD are server side OS's and need to be stable and reliable. Therefore, anyone who has a server will complain if it has to be restarted daily. Desktop computers on the other hand is shut down frequently, allowing OS's that have a short uptime to exist in the marketplace. The "uptime" in Table 8 is a difficult term to visualize. Short can indicate that computers have to be restarted daily or even weekly. Permanent indicates that the system is up until power failures occur or when maintenance needs to be done. There is no reason for a Linux box to be permanently up, it can be restarted every day as a normal desktop PC would.

Because Windows has nearly 90% of the desktop computer market [18], it is almost obvious to say that it is the best choice, because everyone uses it. Just because 90% of the population uses something does not make it "the best choice". Our requirements for software is low cost. Therefore, Linux or Free BSD would be the better choices in this case. We have a certain amount of functions that needs to be available to the users and therefore, we need to verify that a free OS can be used and achieve full functionality.

Comparing different OS's to their desirable features is also advantageous. This comparison is shown in Table 9.

²⁰ This depends mainly on the version of Windows. Windows NT is also a server side OS.

Table 9: OS comparison against desirable features

OS	Desirable Feature	Criticism
Dos	Efficiency	Compiled by Microsoft for one specific architecture to be compatible with all x86 systems.
	Reliability	The same input generates the same output.
	Maintainability	Dos is not in use any more, and Microsoft has to bring out updates, thus DOS is not maintainable.
	Size	Less than 10Mb disk size was needed.
Windows	Efficiency	Compiled by Microsoft for x86 architecture. Windows is compiled for compatibility in stead of optimal use for all processors.
	Reliability	Microsoft tries to guarantee full compatibility and reliability.
	Maintainability	Microsoft has to bring out updates, thus Windows is not maintainable by users. Microsoft maintains Windows with service packs, this is only effective for people connected to the Internet.
	Size	Typically 200Mb and up depending on user applications.
Linux	Efficiency	Default architectures include i386, i486, i586/i686, PPC, SPARC and x86-64bit. Can be compiled by the user for any processor available today.
	Reliability	Linux is stable and used/tested on high performance server systems.
	Maintainability	Patches freely available and package updates available almost daily on the Internet. Updates available on CDROM almost twice annually depending on the distribution.
	Size	Distributions from 8Mb and up but they are highly customizable [25]. Distributed with X from 100Mb and up, depending on user applications.

2.5.2. GNU Public License

A large range of operating systems exist in the world today. Commercial OS's are those you pay for to obtain and use. Open Source projects are programs which can be used with a GNU Public License (GPL). Open source projects

create user applications and OS's. These free applications and OS's are suppressed by commercial OS companies and software patents. As a result of this suppression GPL OS developers are not given access to hardware specifications for proper device driver development. This would force most people to think that free OS's would not support their hardware, which is true in some cases. In general most hardware are already supported by open source projects without the user installing device drivers.

2.5.2.1. What does open source mean?

The primary goal of open source is to make the source code of various commonly used UNIX programs available. Source code holds many advantages over normal binary files [18]. Source code can be compiled for specific processors which means better efficiency or performance. When people use open source they will likely improve on the source. The open source agreement only asks that the improvements be shared with the rest of the community. This allows open source to improve at an astonishing rate. In contrast to closed binary executables where only a small number of programmers try to improve on the source code (which they keep well hidden) while the people who pay for those programs are left in the dark [18].

Other goals are also defined by OSI (Open Source Initiative) in its Open Source Definition. Most of the following rules for acceptable open source licenses are to protect the freedom and integrity of the open source code [18]:

- Free Distribution – An open source license can't require a fee from anyone who resells the software
- Source code – The source code has to be included with the software and not be restricted from being redistributed.
- Derived works – The license must allow modification and redistribution of the code under the same terms.
- Integrity of the authors source code – The license may require that those who use the source code remove the original project's name or version if they change the source code.

- No discrimination against persons or groups – The license must allow all people to be equally eligible to use the source code.
- No discrimination against fields of endeavour – The license can't restrict a project from using the source code because it is commercial or because it is associated with a field of endeavour that the software provider doesn't like.
- Distribution of license – No additional license should be needed to use and redistribute the software.
- License must not be specific to a product – The license can't restrict the source code to a particular software distribution
- License must not restrict other software – The license can't prevent someone from including the open source software on the same medium as non-open source software
- License must be technology-neutral – The license can't restrict methods in which the source code can be restricted.

2.5.3. Assessment of the open source agreement

Can Open source be used in this product? We want to sell these multimedia readers as a complete unit to the students. Without the open source software on this device, it is useless. If we cannot use open source for this project because we resell it, then commercial software will increase the total cost.

Free distribution requires that a fee cannot be required from anyone who resells the software. Therefore, if we resell the hardware together with the open source software (whether we make a profit or not), then the original writers of the source code cannot claim a fee for its sale.

Source code requires that we distribute the source code with this product. It does not state how the source code needs to be distributed. According to the point that states that the “**License must be technology neutral**” we may make a CDROM disk containing the code to distribute together with our media reader. The on-board memory need not hold the source code and can therefore, be smaller.

We do not intend to change the source code in any way. Therefore, the following points do not apply to this project:

- **Derived works**
- **Integrity of the authors source code**

No discrimination requires that the original writer may not discriminate against fields of endeavour, groups, or persons. Therefore, the original writers may not stop our use of the source code because we intend to resell the multimedia reader as a whole.

Distribution of license requires that our product be handed down with the same license. This implies that the terms and conditions for the open source applications do not change when the multimedia reader is sold to the students.

2.5.3.1. Conclusion

We may resell the product as a whole. There are some important guidelines to keep in mind:

- We must include the source code with the readers. Whether it is on the reader itself or on a separate CDROM disk.
- We may not discriminate against anyone who wants to buy this multimedia reader. Therefore, this product may not be exclusive to students!
- We may not prevent the copying of programs on this multimedia reader in any way.
- We may therefore, not patent or copyright the final product.

2.6. Applications

Considering that the cost needs to be kept as low as possible, open source programs would be good contenders for the final product. Only a few open source programs will be discussed in this section because there exists a too large a number of available software to mention all of them. There are enough open source projects available to suit any user's specific desire. In the future

when the user feels he/she knows enough of this multimedia reader then he/she may change the software to what he/she feels more comfortable with.

The following section will now look at possible user programs to run in the environment created by the OS. Firstly there will be study material which consists of text, sound and video in a standard format. Therefore, we may use a web browser which is capable of displaying multimedia. Web browsers use a standard format known as HTML (Hyper Text Mark-up Language). A graphical web browser needs a graphical user interface or window manager. A window manager needs the X protocol. Secondly there must be a writing tool or word processor for students to compose their own notes. The word processor needs a window manager which needs X.

This project will use the following features/applications:

1. Graphical user interface
 - 1.1. X
 - 1.2. Window manager
2. Web browsers
3. Sound
4. Video
5. Writing tools

2.6.1. X

The graphical part of all computing systems are handled as a normal program. The GUI is handled by the X Window System. "The X Window System" is a protocol laid down by the X-Consortium. Any program that conforms to this protocol and which meets the criteria laid down by the X-Consortium can be called X [17].

The X Protocol was developed in the mid 1980's and was intended to provide a network transparent GUI for the UNIX operating system. X provides for the display and management of graphical information. The X Protocol distributes

the processing of applications by specifying a client-server relationship at the application level. The benefit is true client-server and distributed processing [34].

The advantages of this approach are many [34]:

- Local and network based computing look and feel the same to both the user and the developer.
- The X server is highly portable allowing support for a variety of languages and operating systems.
- X clients also have a high degree of portability.
- X can support any byte stream oriented network protocol, local or remote.
- Applications do not suffer a performance penalty.

“The X Window system” will be used to create the interactive graphical environment. Graphical environments make it easier for non-technical users to use the system effectively. The look and feel of the project is improved greatly by the graphical environment together with the window manager. The window manager is another program that uses X in order to function correctly.

Different X Window Systems exist in the open source world, they are Xorg [34] and XFree86 [35].

Xorg and XFree86 are both freely re-distributable open-source implementation of the X Window System. This application provides a client/server interface between display hardware (the mouse, keyboard, and video displays) and the desktop environment, while also providing both the windowing infrastructure and a standardized application interface (API) [34], [35].

2.6.2. Window managers

The window manager is a software program that presents the user with a graphical interface to the multimedia reader. The window manager handles the commands given by the user such as opening programs with a click of the mouse. The window manager also helps the user to arrange the open programs in an orderly fashion.

2.6.2.1. Fluxbox [19]

Fluxbox is a small and light window manager for X. Fluxbox can handle styles, colours, window placement and themes [19].

Fluxbox is a very simple and light window manager that uses up roughly 15Mb of disk space. Fluxbox has very little dependencies therefore, it can be installed without first installing a lot of other system libraries.

2.6.2.2. XFCE

XFCE is a fast and lightweight desktop environment that is both visually appealing and easy to use. XFCE loads programs faster and more efficient than most other window managers. This window manager embodies the traditional UNIX philosophy of modularity and re-usability. XFCE adheres to the standards set by freedesktop.org. XFCE is customizable with downloadable plug-ins, icons and themes [37].

XFCE is a small application covering roughly 40Mb of disk space.

2.6.2.3. KDE

KDE is a network transparent contemporary desktop environment for UNIX workstations. KDE seeks to fill the need for an easy to use desktop for Unix workstations, similar to the desktop environments found under the MacOS or Microsoft Windows [38].

Together with free implementations of UNIX (such as various flavours of Linux), KDE creates a completely free and open computing platform available to anyone free of charge including its source code for anyone to modify [38].

A vast amount of applications depending on KDE exist in the open source community [38], [39]. These high quality programs are shared between users and maintained by their creators and users.

KDE is a much larger window manager that is distributed on a CD and takes up much more disk space than XFCE and Fluxbox. KDE can cover up to 1Gb of disk space depending on which packages are installed with it.

2.6.3. Browsers

Many browsers exist in the world today where the stability and standard features or available plug-ins determines the success of the browser.

2.6.3.1. Mozilla

The Mozilla project was launched by Netscape in 1998 [42].

Mozilla browsers include features such as [42]:

- **Tabbed browsing** gives you a better way to surf the net. You no longer have to open one page at a time. With tabbed browsing, open several pages at once with one click. And now your homepage can be multiple tabbed pages.
- **Pop-up blocker** lets you surf the web without intrusion. Advanced popup blocker notifies you when pop-ups are blocked. You can also block pop-ups on a site per site basis.
- **Image Manager** lets you block images to block offensive images or to speed up the rendering of web sites.
- **Find as you type** gives you another way to navigate a page. Just start typing to jump from link to link or to find a word or phrase within a page.

Plus all the features a modern browser should have including: Advanced security settings; Password, Download, and Cookie managers; Themes; multi-language and multi-platform support; and, the latest in Web Standards [42].

Mozilla is a large browser consuming up to 60Mb of disk space.

2.6.3.2. Mozilla-Firefox

Firefox is a new generation, state of the art open source browser. Firefox is small, free and runs with minimal resources, perfect for use in this project.

Firefox, previously known as Firebird, is a stripped down, streamlined rebuild of Mozilla. Firefox still supports the tabbed browsing. Pop-up ads are also easily blocked [22].

Microsoft's Explorer may indeed still command the vast majority of the Web browser market. But the "Browser Battle" is on again [22].

Mozilla Firefox consumes up to 20Mb of disk space but does not include all plug-ins available to web browsers.

2.6.3.3. Netscape

Netscape includes features such as [43]:

- Netscape warns you if you come across possible spyware sites, and automatically configures its settings to help protect you.
- Tabbed browsing has been expanded so you can open all of your favorite web sites in one window automatically.
- New Passcard Manager remembers all your screen names and passwords and can auto-fill that information on sites.

Netscape consumes up to 60Mb of disk space.

2.6.4. Sound programs

Sound programs are user applications that play music files while the user is free to work with other applications.

2.6.4.1. XMMS

XMMS (X Multi Media System) is a small and free music application that resembles Winamp.

XMMS has been voted the favourite audio tool for the past five years by the readers of LinuxJournal [24]. XMMS supports various sound input plug-ins such as WAV, MP3, FLAC, Audio CD's and OGG (the free alternative to MP3's) to name but a few. Various output plug-ins are also supported such as esound, aRts, ALSA and OSS [24]. Other plug-ins are available on XMMS' homepage [24].

2.6.4.2. BMP

This media player is an upgraded split version of XMMS. BMP started as Beep media player, but BMP does not necessarily stand for anything today. BMP is based on GTK2 where XMMS is based on GTK1.2 (which is deprecated²¹). Although BMP is far from finished with respect to its main goals [40], it is a fully functional music player with a promise of upgrades in the future.

2.6.4.3. JUK

JUK is a luxury background music player which puts an icon in the taskbar while active.

JUK is part of KDE and distributed with all current Linux and many UNIX distributions [41].

JUK presents the user with an easy to use play list and meta data. The play list has many functions such as a collection list, search capabilities, queuing

²¹ Depreciated meaning that the software is old and not widely used today.

capabilities and folder play list. The player itself can interpret many file formats such as MP3, OGG, FLAC and more. JUK has output plug-ins such as aRts²², Gstreamer and aKode [41].

2.6.5. Video players

2.6.5.1. Xine [44]

Xine is a free multimedia player. It plays back CDs, DVDs, and VCDs. It also decodes multimedia files like AVI, MOV, WMV, and MP3 from local disk drives, and displays multimedia streamed over the Internet. It interprets many of the most common multimedia formats available - and some of the most uncommon formats, too .

Xine features include:

- **xine-engine**

The core of xine is responsible for synchronizing audio, video and overlays. It provides high performance communication functionality between modules, logging capability, unified configuration system, On Screen Display support, fast MMX/MMXEXT/SSE memory transfers, among other important things.

- **input plug-ins**

They provide input data to xine and may be seen as an abstraction layer for DVDs, files, http, pipes, VCDs so that the rest of xine does not need to know from where the data is coming. Several input plug-ins for xine are already available over the Internet from third party programmers.

- **demuxer plug-ins**

These plug-ins are responsible for understanding file formats like AVI, MOV, ASF and others. These file formats don't define how video and audio are encoded but how the encoded data is mixed together (multiplexed). An AVI file, for example, may have DivX4 video and MP3 audio, MPEG2 video and AC3 audio, etc.

- **decoder plug-ins**

These plug-ins receive the encoded data (video, audio or subtitles) and

²² aRts is KDE's default sound manager system.

deliver them uncompressed to the engine to be played or shown. Examples of encoding formats (also called "codecs") are MPEG2, MP3, Ogg Vorbis and Windows Media Video 7/8/9.

- **output plug-ins**

xine runs on a wide variety of hardware and operating systems therefore, it need to have different means of displaying video and audio. These plug-ins are like drivers, they talk directly to the system so xine-engine does not need to handle the details. Some video output plug-ins have been developed to utilize several hardware capabilities like colour conversion, scaling and refresh sync to provide the best multimedia experience and at the same time, requiring less CPU processing.

- **post effect plug-ins**

These can be use to apply arbitrary effects to video and audio before they are sent to output. Examples include visual plug-ins like GOOM (generates colourful images from the music being played), audio filters like echo, equalizer or even a video picture-in-picture plug-in playing more than one stream at the same time.

2.6.5.2. MPlayer [45]

MPlayer is a free movie player which runs on many systems. It plays most MPEG/VOB, AVI, Ogg/OGM, VIVO, ASF/WMA/WMV, QT/MOV/MP4, RealMedia, Matroska, NUT, NuppelVideo, FLI, YUV4MPEG, FILM, RoQ, PVA files, supported by many native, XAnim, and Win32 DLL codecs. You can watch VideoCD, SVCD, DVD, 3ivx, DivX 3/4/5 and WMV movies.

MPlayer is a customizable player that supports a wide range of special features found in hardware today. MPlayer can detect processor features such as MMX, SSE, 3DNow and many others. These features gets used by the software to lessen the processor load while playing video files.

2.6.6. Writing tools

2.6.6.1. *AbiWord* [23]

AbiWord Personal is a lean and fast full-featured word processor. It can import many formats including Word97/2000 and RTF, Palm, Psion, DocBook and XHTML documents, and can export to RTF, Palm, Psion, XHTML, Text, and LaTeX formats. Linux Journal calls it "an elegant, open source word processor that delivers the Word functionality most people use."

AbiWord is a very basic, low requirement word processor which resembles Microsoft Word.

AbiWord is a free GNU word processor which has attracted lots of attention in the past few years. It has even reached TechTV's download of the day April 20th 2004.

"This totally free package is packed with tons of features that should make MS Word jealous, though there are some features that is still missing in the newest release such as a grammar checker" [23].

2.6.6.2. *Open Office* [47]

Open Office is both a multi-platform and multi-lingual office suite and an open source project. Compatible with all other major office suites, the product is free to download, use, and distribute.

Open Office offers different components such as:

- Writer – equivalent to MS Word
- Calc – equivalent to MS EXCEL
- Impress – equivalent to MS Presentations
- and more

2.6.6.3. Kate (KDE Advanced Text Editor) [46]

Kate is not a word processor such as ABI Word or Open Office. Kate is a multi document editor native to KDE. Kate is part of KDE which is free therefore, Kate is free software.

Kate has features such as:

- Edit all kinds of text files even if they are BIG (i.e. open a 50MB file in a few seconds).
- Powerful syntax highlighting engine, extensible via XML files.
- Code Folding capabilities for C++, C, PHP, ...
- Dynamic Word Wrap - long lines are wrapped at the window border on the fly for better overview.
- Multiple views allows you to view more instances of the same document and/or more documents at one time.
- Can reopen files if you want it to.
- User can choose the encoding he wants for each file at save/open dialog.
- Built in dockable terminal emulation.
- Sidebars displays a list of open documents, a directory viewer with a directory chooser, a filter chooser and more.
- A plug-in interface to allow third party plug-ins.
- Default project handling, possibly to be overridden by project handling plug-ins.
- A "Filter" command allows you to run selected text through a shell command.
- Global grep dialog

3. Chapter Three – Feasibility study

3.1. Introduction

There exist two main contenders for the OS namely Windows (“not free”, yet well known²³) and Linux (presumably free, yet less people are aware of its existence [18]).

Consider a computer that costs R3000 with more powerful features than a computer that costs R2000. If a certain software package costs R1000 less than its contenders' software package, but the cheaper software can only produce the same results on the stronger computer than its contender can produce on the cheaper computer, then the two packages cost roughly the same in the end.

This feasibility study will aim to determine if Linux can be used to lower the cost of the multimedia reader as a whole. This cannot be done without a benchmark therefore, Windows will be the benchmark. Windows will be the perfect candidate because it holds the current monopoly with roughly 90% of the desktop market [18]. Finally there will be a discussion on how the combinations of hardware and software produces different price implications. In other words, we consider the play-off between:

1. Hardware cost vs. OS functionality and
2. OS cost vs. System functionality

To start the feasibility study, we must consider a few contenders for the hardware and the software.

²³Windows is used on roughly 90% of desktop computers [18].

On the hardware side we have chosen three different processor boards or SBC (single board computers). A SBC is a stand alone computer that is very small compared with a desktop computer. SBC's have embedded functionalities that take up very little physical space but can produce the same results as desktop computers.

Software can be very OS specific and can generally not be chosen without considering the OS first. For this project we need a low cost OS. Because Linux is free it will be used if it can produce acceptable results with the given hardware. If stronger or more expensive hardware is needed to produce acceptable results, then in effect Linux starts to cost money. Windows is well known and may serve to be an excellent benchmark. Most hardware vendors create working Windows drivers for their products giving Windows the apparent advantage.

This chapter will present a feasibility study on the new development to determine whether Linux can be used to cut the cost of the system or if Windows will ultimately be cheaper. Factors such as installation size and mean CPU usage were not considered at this stage, but rather achieving the full functionality of the entire system. The next chapter will focus on optimizing the software by using specially selected programs and other possible methods. Optimization of the software includes not only cutting down on the total installation size of the software but also the performance of the software on the specific hardware.

Microsoft Windows and Linux were implemented on each SBC. Each OS was installed with the maximum number of options available on the installation media because only full functionality was important for the feasibility study. Directly after a clean installation (no extra software) each OS was tested against the required functionality. Then the functionality achieved with Linux was compared to that achieved with Windows on the same hardware. After the comparison of each OS with the specifications, attempts were made to correct

the problems presented by each OS after a clean installation. Again each OS was tested against the required functionality and Linux was compared to Windows. Linux must be compared to Windows because we are measuring the feasibility of the use of Linux in this specific implementation.

Therefore, this feasibility study will be successful if the requirements can be met with a hardware and software combination that satisfies the cost requirements.

It was stated in the previous chapter that it is hard to determine which functionality is part of the OS and where user applications start. They all build on each other. The OS cannot enable features that are not supported by the hardware. The user applications cannot enable features that are not supported by both the hardware and the OS. This information seems abstract because the user will mostly interact with user applications and never with the OS itself. This is because the main task of the OS is to create a environment for other programs to work in [16].

Required OS functionality include:

- CDROM as media input;
- Sound Output and
- Video capabilities.

Required user applications include:

- Display web pages and
- Create notes.

Finally we will conclude on the suitability of each installation.

3.2. Software

The two OS's to implement are Windows and Linux. This project uses Linux to cut the total cost of the system therefore, Linux will be discussed more thoroughly. There are a wide range of versions of Windows available. All of

these versions are not fully compatible and sometimes confusing when upgrading from one version to the next. With Linux the versions stay fully compatible and are configured in exactly the same way as previous versions of the same distribution. There are huge differences when switching from one distribution (or “flavour of Linux”) to another but the internal working and configuration-basics stay the same between versions.

Choosing a version of Windows is therefore, just as difficult a task as choosing a specific flavour of Linux. Flavours of Linux are different in order to suit the users who use them, it depends a lot on the users' personality.

3.2.1. Windows 98

Windows 98 was chosen because the processor boards are fairly slow in comparison with standard desktop computers that exist today. Windows 98 was built on machines that were available in the 90's which is slow compared with current computers but roughly in the same range as our SBCs. Windows XP is built on newer technology and runs well on today's computers.

Windows 98 costs less than Windows XP [36]. This multimedia reader will eventually be used for home purposes with only basic functions. Windows 98 is better than Windows XP for running older type applications [36], we can accept older type applications if they are functional. Windows 98 is easier to use than Windows XP [36].

Windows XP is built on NT technology [36], but the feasibility need only prove that the hardware works.

Table 10: Minimum system requirements for Windows [48]

Specs	Win 98	Win ME	Win XP	Win CE	Win 2000
RAM	32	32	128	16	64
CPU	300	300	300	N/A	300
Disk Space	300Mb	300Mb	1.5Gb	32	1.5Gb

3.2.1.2. Windows 98 installation suite²⁴

- Windows' Graphical user interface (Window manager)
- Internet Explorer
- Microsoft Network
- Media Player
- Sound recorder
- Wordpad
- Games (important learning experience)

3.2.2. Slackware Linux

Slackware is one of the oldest Linux distributions and works essentially the same as all the other Linux distributions available. Slackware offers the following advantages over other available distributions [18]:

3.2.2.1. Better Comprehension of the OS

Text based commands and configuration helps the user to learn more about how Linux works. GUI tools often hide the complexity and unfortunately limits the features that users can use. If something goes wrong, it can be very hard to debug with most graphical interfaces.

3.2.2.2. Less overhead of system resources

Graphical user interfaces consume far more resources than their command-line counterparts. Slackware relies primarily on basic Linux commands, text-based configuration files and some simple menu-driven administration tools.

3.2.2.3. Better for low-end computers

Slackware can be installed on a i386 with as little as 4Mb of RAM. If you want a GUI, the installation procedure lets the user choose small, efficient window

²⁴ The list of programs that come with Windows 98 is not complete, only the programs that were directly used in the feasibility study are explicitly stated.

managers, Web browsers, mail clients and other graphical tools. Luxury window managers such as KDE and GNOME are also available.

3.2.2.4. Unaltered packages

Slackware doesn't mould the software it includes into one look-and-feel. All packages installed are exactly as the original project developers intended them.

For this feasibility study some drivers will have to be compiled. Therefore, we need a distribution with an acceptable compiler. Slackware is a distribution that is mostly used for development [18].

Slackware does install to 2Gb if a full desktop install is done. However size is not important for the feasibility study. Only the final product must be small enough to fit onto the processor boards' memory.

Slackware Linux is free and available from the Internet [27].

3.2.2.5. Slackware 10.0 installation suite

- **Base** - The base Linux system needed to log in and run programs
- **Applications that do not need X** - Various useful tools and applications for everyday use
- **Libraries** - Base C libraries and dynamically linked libraries needed to run various programs
- **Development** - Compiler tools and libraries
- **Network** - Network tools such as web browsers, mail clients, server/client side software
- **X** - The binaries of Xorg and development source code
- **Applications that need X** - Window managers and other programs that exhibit Graphical User Interfaces

Slackware 10.0 Version info:

- Size: 2.5Gb
- Kernel: 2.4.26
- Glibc: 2.3.2
- Architecture: i386/i486
- ALSA²⁵ support: Yes (version 1.0.5)

3.3. Hardware

The hardware was chosen with the following characteristics in mind:

1. Availability in South Africa
2. Cost
3. Minimum requirements for the OS's to be implemented
4. Minimal unused hardware
5. Ease of use in the final product

We wanted different SBCs that could be tested to determine the minimal required hardware resources to produce the required results. Reading the minimal system requirements to run software is helpful but the people who set those requirements had another functional system in mind than our multimedia reader.

For example: An OS may run very smooth on a 16MHz i486 processor with 8Mb RAM, but the creators never planned to use a graphical user interface which takes up much more resources than the minimum.

²⁵Advanced Linux Sound Architecture is a new development in Linux sound support and is embedded in the 2.6 Linux kernel. ALSA can also be used with the 2.4 kernel but it is not yet native to the 2.4 Linux kernel.

3.4. NC546

The first processor board chosen was the NC546. This board had all the necessary features such as CPU, sound, storage and media output. This SBC also had some (initially) unneeded features such as a network interface and USB. The NC546 board was within the price range at a cost of R1500 excluding VAT and the various cables for different connectible peripherals.

Table 11: On board features of the NC546

Feature	Description
CPU	National Semiconductor GEODE GX1 300MHz MMX
Sound	National Semiconductor AC97
Display Card	Media GX 4Mb shared memory
Network Interface	Realtek 8139C 10/100 Mbps
Storage	40Gb Hard disk 24x CDROM (5v)
Memory	128Mb (4Mb shared)
USB	Version 1.1 (12 Mbps)
Price	R 1,500.00

As seen in Table 11 the NC546 does satisfy the requirements stated earlier. The hardware must be able to reach the goal because software only enables the hardware's abilities, software cannot enable abilities which is not built into the hardware. The feasibility study now aims to produce a working system that can reach its goal with two different OS's. The first OS to be implemented is Windows.

3.4.2. Windows

Windows was installed and tested against the requirements. The CDROM was functional, Internet explorer opened test pages that displayed text, pictures and graphs. Notes were made with Wordpad and saved on the storage media without difficulty. Video files could not be played on the system. Sound played but with no actual output to the speakers. A summary is available in Table 12.

Table 12: Windows on NC546 before drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Low quality
Display Video	No
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

Windows does manage to quickly set up a computer for normal everyday use. The default settings that installs with MS Windows seem to be chosen well for the common desktop computer. Windows does support CDROM, hard disks and low quality video output. The video output can easily be corrected with drivers that helps MS Windows to effectively use the video hardware.

3.4.2.2. Problems before additional software was installed

After a full installation of Windows, a few problems surfaced:

1. Video was unacceptable
 - 1.1. Resolution 640x480
 - 1.2. Video was very slow
 - 1.3. Video was 16 colours
2. Sound did not work

We discuss each problem that surfaced separately and the steps taken to resolve them.

1. Video:

The display was 16 colour at 640x480 resolution. 16 Colour displays are able to display colour but only with discretion between 16 colours in contrast to most modest 16bit displays available today that can generate up to 65 thousand colours. The resolution of 640x480 also creates a problem because the detail that can be displayed on such a screen is very limited.

The display rendering was not nearly fast enough to watch video clips. The system tried to open video clips but exited with errors.

The video driver provided by the manufacturers was installed and detected the new hardware:

- Media GX device
- New Plug-and-Play monitor

After the system restarted, the video card was functional. This fixed the problem with slow video and the resolution. The resolution could now be set to 800x600, 16bit. The display was fast enough to watch MPEG2 files.

The system could not play DivX which is MPEG4 and needs a stronger processor. Various versions of the DivX codec was installed but to no avail. The processor was too slow. This problem cannot be resolved with drivers because the physical hardware cannot handle the decompression fast enough.

2. Sound:

There were no actual sound output to the speakers. First we checked the volume to see if the sound was not muted. The problem was that no sound driver was available to Windows.

The sound driver provided by the manufacturers was installed. Windows then claimed to have found new devices:

- National Semiconductor Audio codec
- National Semiconductor Midi Device
- National Game port

After all the drivers were installed the sound was still not functional. The Windows device manager indicated the following error message:

Device is enabled but inactive due to an unknown problem.

Conclusion: The sound driver did not work, or the hardware was faulty.

A new driver downloaded from www.driverguide.com was installed. After the same procedure as the original driver and two system restarts Windows persisted with: "Device is enabled but inactive due to an unknown problem".

Conclusion on the sound card

Two different drivers were installed and none worked. Therefore, we can conclude that the drivers were not compatible or that the sound hardware was not functional.

Windows' performance measured against the specifications after drivers were installed can be found in Table 13.

Table 13: Windows on NC546 after drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	Yes
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.4.2.3. Conclusion

Except for the sound card the whole system was functional within performance limits. Hardware problems persisted because of the CPU that was not fast enough to watch heavily compressed video, which cannot directly²⁶ be fixed with software.

3.4.3. Linux

Slackware Linux 10.0 was installed with no problems during the installation procedure. Everything was installed onto two partitions one large partition mounted on root and one 256Mb swap partition.

After the installation the same tests were conducted as with Windows. The CDROM was functional, test pages were opened with the Mozilla browser and all data were displayed without any problems. KDE (the window manager) complained that there were no sound hardware detected and sound output will be disabled. Video files were opened and the video was of acceptable quality. Notes were made with ABI-Word and saved on the storage media.

²⁶ In the next chapter we will see that software can be used to correct minor problems that exist due to slower processors. "Minor corrections" - meaning we cannot correct everything although improvements of around 30% better performance is not uncommon.

Table 14: Linux on NC546 before drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	Yes
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

A summary of the Linux installation on the NC546 SBC is available in Table 14. Everything worked fine save the sound. Drivers for the sound card were not available in the generic Linux kernel and will have to be installed separately.

3.4.3.2. Problems before additional software was installed

After a full installation of Linux, one problem surfaced:

1. Sound did not work

1. Sound:

The drivers provided by the manufacturers for the sound card were unpacked but did not compile. During compilation the compiler gave many errors such as incorrect pointer declaration and code breaking strict aliasing rules.

The drivers provided by the manufacturers were not usable. Therefore, we may use commercial drivers for the feasibility study. The commercial drivers are written and sold as working drivers therefore, these drivers must work.

The commercial third party driver from 4Front was downloaded and installed with a three month trial license. The OSS driver installed with a simple script and seemed to be in perfect working condition. The driver picked up the sound card correctly, however when the driver tested the hardware, the loop back

tests failed when the left channel was being tested. This disabled the commercial OSS driver completely and the sound was still not functional.

Finally after all the sound drivers were tested, none could produce any sound with the specific hardware.

Linux' performance measured against the specifications after drivers were installed on the system is visible in Table 14.

3.4.3.3. Conclusion

MPEG4 files could initially not be played on this system. We accepted that the processor was too slow to play MPEG4 compressed files because Windows had the same problem. To prove that the processor was too slow and not being wrongfully accused we optimized this instance solely for being sure that we made the correct assumption. More complete optimizations is done in the next chapter. This is because we have to know if 300MHz is below the minimum requirements for the multimedia reader.

Table 15: Linux on NC546 after drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	Yes
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

Table 15 shows the summary of the requirements satisfied by the Linux/NC546 combination.

However after the Linux kernel was recompiled for i586/i686 and Xine (the free video player) was reinstalled from source code to support MMX Linux did play MPEG4 files. Xine did however give errors stating that the system was too slow to play the media – the number of dropped frames was too high. Although many frames were dropped, according to the media player, the video quality was acceptable. This phenomena suggests that software can be altered to produce better results with the same hardware. This approach is not applicable to Windows because Windows is not distributed with its source code or its compilers.

Except for the sound output everything seemed to be in perfect working condition.

3.4.4. Conclusion of NC546

At first sight Linux seemed to have better hardware support because Windows did not support the display card by default and Linux did. After drivers were installed for the display card Windows did support the display card. Windows did not support the network card by default and Linux did. Again network driver support were available for Windows. Neither Windows nor Linux could play sounds through the sound card before or after drivers were installed therefore, we may conclude that the hardware may have been faulty.

Both Windows and Linux were fast enough to play MPEG2 videos. Neither OS could play MPEG4 which has stronger compression, however Linux did play the files with warnings that the system was too slow to play the files effectively. This proves that the 300MHz processor is below the minimum system requirements for video output.

At this stage of the feasibility study, Linux seems to have the upper hand on support while both seem to run at the same speed. However Linux could be easily optimized to work notably faster than Windows **at no extra cost**. For this processor board Linux is indeed cheaper than Windows.

This processor board cannot be used for the final product because it is too slow and not all features were working. If MPEG4 compression is to be used in the future then this processor board will not be able to produce acceptable results. The physical properties of this processor board makes it hard to use in the final project because it is large and does not run directly from battery.

3.5. ePanel MGX

The second processor board chosen was the ePanel MGX. The ePanel MGX has an upgraded version available in large quantities. Therefore, this board was chosen, because if we can produce results with this board then the upgrade will take care of any speed problems we have with this SBC.

This board had essentially the same features as the NC546 but with a slower processor. This SBC was not within the price range, but still tested for the **feasibility study** for the mentioned reasons. Further differences were that this SBC had an on-board charger module and a direct battery input which both the other two boards lack. Furthermore this SBC was designed to be mounted back-to-back with a LCD screen.

Table 16: On board features of the ePanel MGX

Feature	Description
CPU	National Semiconductor GEODE GX1 266MHz MMX
Sound	Geode AC97
Display Card	National Semiconductor 4Mb shared memory
Network Interface	Realtek 8139C 10/100 Mbps
Storage	40Gb Hard disk 24x CDROM (5v)
Memory	128Mb (4Mb shared)
USB	Version 1.1 (12 Mbps)
Price	R 4,500.00

Table 16 shows that the hardware is capable of all the features listed in the requirements earlier in this chapter. One problem may be the processor speed because the previous SBC (NC546) could not play MPEG4 (heavily compressed) video and this SBC has a slightly slower CPU clock speed.

3.5.2. Windows

The same version of Windows was installed on this SBC as on the previous board. Therefore, we expected the same problems as with the first board because Windows' native driver support stays the same.

The tests conducted with this SBC were the same tests conducted with the previous SBC running Windows.

The results of the tests are shown in Table 17. MS Windows was consistent with the supported hardware and proved to set up the SBC with well chosen initial settings.

Table 17: Windows on ePanel MGX before drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Low quality
Display Video	No
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.5.2.2. Problems before additional software were installed

After a full installation of Windows, a few problems surfaced:

1. Video was unacceptable

- 1.1. Resolution 640x480
- 1.2. Video was very slow
- 1.3. Video was 16 colours
2. Sound did not work
3. Network did not work

1. Video:

The display showed the same symptoms as with the NC546 SBC. The display rendered only 16 colours and the resolution was 640x480 which prevented much detail to be shown in the pictures and graphs during the tests. The problem was corrected by installing the driver, but the results changed.

After the new device drivers were installed the video card was functional. This fixed the problem with the resolution. The resolution could now be set to 800x600, 16bit. However the display was still not fast enough to watch MPEG2 files.

Conclusion on the video:

There exists two possible causes for the dysfunctional video output. The processor is too slow, or the display cards' rendering speed is too slow. The display card is connected through a PCI²⁷ bus and not AGP²⁸ which is newer and much faster.

2. Sound:

The sound card drivers were installed to no avail. The sound card was still not functional.

The Windows device manager indicated the following error:

Device is enabled but inactive due to an unknown problem.

²⁷ Peripheral Component Interconnect (PCI) is a bus used to transfer data between various peripherals and the CPU.

²⁸ Accelerated Graphics Port (AGP) is a special bus used to connect video output hardware with the CPU. AGP was developed to advance the available bandwidth used by graphics in computers.

Conclusion: The sound driver did not work, or the hardware was faulty.

A new driver downloaded from www.driverguide.com was installed. After the same procedure as the original driver and two system restarts Windows persisted with: "Device is enabled but inactive due to an unknown problem".

Conclusion on the sound card

Finally we may conclude that the sound card drivers were not compatible or that the hardware was faulty. This is the second card with almost the same sound card as the previous NC546, the chance that both had hardware issues would be very slim.

3. Network

Network drivers were provided by the manufacturers. After the network was installed the network was fully functional.

Windows had the following performance measured against the specifications:

Table 18: Windows on ePanel MGX after drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	No
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.5.2.3. Conclusion

Except for the sound and video everything seemed to work well. The sound card may have a hardware problem or the driver may be incompatible with

some part of the system. The processors' speed is slower than the previous SBC. Therefore, we may conclude that the processors' speed may be responsible for the dysfunctional video.

3.5.3. Linux

Slackware Linux 10.0 was installed with no problems during the installation procedure. Everything was installed onto two partitions; one large partition mounted on root and one 256Mb swap partition. Note that installing everything onto one large partition is generally not a good configuration for many reasons, however this is only the feasibility study and the number of partitions do not hinder the functionality of this system.

The tests that were run with Linux on this SBC were the same tests described with the NC546 running Linux. Because of the slightly slower processor we expected a slower system which could not play video's.

Table 19: Linux on ePanel MGX before drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	No
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.5.3.2. Problems before additional software was installed

After a full installation of Linux, two problems surfaced:

1. Sound did not work
2. Video was slow

After searching the Internet for Geode sound drivers the following were found:

1. Commercial OSS driver from 4Front
2. Geode OSS driver
3. Geode ALSA driver version 0.5.12b

1. Sound:

1.1. Commercial Driver from 4Front:

The commercial OSS driver from 4Front was downloaded and installed with a three month trial license. The OSS driver installed with a simple script and seemed to be in perfect working condition. The driver detected the sound card correctly, however when the driver tested the hardware, the loop back tests failed when the left channel was being tested. This disabled the commercial OSS driver completely and the sound was still not functional.

1.2. OSS Driver from the manufacturers:

The OSS drivers did not compile because of the Linux kernel version. Drivers depended on kernel 2.4.17, while 2.4.26 was installed. We therefore, downgraded the Linux kernel to version 2.4.17. After the 2.4.17 kernel was installed, and various recommended patches were applied, the drivers compiled without problems. The drivers did not initialize the sound card and returned the following error:

“ERROR: Cannot modprobe geodeoss.o - no such device. Insmod failed”

This error message means that the sound card drivers did not load because the intended hardware was not found on the system.

1.3. ALSA Driver from the manufacturers:

The ALSA drivers did not compile at first because there was no ISA-bus detected on the SBC itself. This problem would be significant if the sound card was connected via ISA, but the card was connected via PCI. There were also problems while compiling drivers for a card named “AD1819” which is in fact an ISA card but does not exist on this hardware. After giving the following special configuration switches to the compiler it compiled without a problem.

```
./configure --with-cards=geode --with-isapnp=no
```

The kernel modules (drivers) loaded with the following warning:

```
no license found: snd-geode.o will taint the kernel
snd-geode.o loaded with warnings.
```

The warning means that if there is a malfunction due to this driver, it is not the fault of Linux or the Linux kernel and the user should be aware of the danger.

The sound card was tested and successfully played a wave file with the following command:

```
aplay sample.wav
```

This demonstrated that the driver worked.

This version of ALSA (0.5.12) was too old to work with the standard GUI software that uses sound cards. This is because ALSA was still being tested when version 0.5 started to surface. Many ALSA functions and system calls changed from the testing phase up to version 0.9. More application interfaces were developed for the newer versions and support for the older versions of ALSA are rare. Therefore, special applications had to be installed to work with this old driver such as "ALSA Player". ALSA Player is able to interface with both ALSA 0.5.x and versions later than 0.9.x.

1.3.1 ALSA Player:

This packages' configuration script did not detect an installed version of ALSA 0.5.x, we do not know the reason for this. After searching through the configuration scripts for quite a while we managed to alter the script enough to enable the linking of ALSA Player with the ALSA driver.

After the following special switches were used to compile the driver it worked with all the standard software.

```
./configure --with-cards=geode --with-isapnp=no --enable-oss
```

The reason is that the driver was now compiled with an OSS capable plug-in. Most of the standard software programs are OSS ready.

2. Video drivers:

Additional drivers for the National Semiconductor display card were installed. The new drivers functioned just as the previously installed drivers did and video was still too slow to watch video clips. The sound of the video clips worked but the pictures were showed at one second intervals (similar to a slide show).

Linux' performance measured against the specifications is shown in Table 20.

Table 20: Linux on ePanel MGX after drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	No
Sound output	Yes
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.5.3.3. Conclusion

Except for the video which we expected to fail everything worked fine. If the video could not play due to a slower processor then optimization might enable the video functionality. Optimization could not enable video functionality.

The reason for the dysfunctional video output may be because of the display cards' rendering speed or the driver. The rendering speed was tested with "glxgears" to perform at an embarrassing 10 frames/second with the NSC driver from Xorg. The VESA driver, recommended by the manufacturers, tested with "glxgears" averaged at around 8 frames/second. Therefore, video cannot be played because the nominal frame rate of video files are 23 frames/second and up.

3.5.4. Conclusion of ePanel MGX

At first sight Linux seemed to have better hardware support because Windows did not support the display card by default and Linux did. After drivers were installed for the display card Windows had support for the display card. Windows did not support the network card by default and Linux did. Again network driver support were available for Windows. Windows could not play sounds through the sound card before or after drivers were installed. Linux could not play sound before drivers were installed but worked nicely after the ALSA driver with its OSS plug-in were installed.

Neither Windows nor Linux was fast enough to play MPEG2 video clips.

At this stage of the feasibility study, Linux does exhibit superior hardware support over Windows 98. For this processor board Linux is indeed better/cheaper than Windows.

This processor board cannot be used for the final product because it is too expensive and too slow. This SBC is suitable for making notes and reading study material that does not include video files. The physical properties of this board makes it easy to package into the final product.

3.6. NC677

The next processor board chosen was the NC677 which has hardware acceleration support for playing DVD. This SBC is half the size of the previous

two processor boards while featuring a more powerful on-board processor and display card. This SBC has a very powerful processor and an AGP graphics card making it easier to display video, than the previous two processor boards. The NC677 has a direct TFT²⁹ LCD output which makes it easy to connect a LCD without the need for more external peripheral converter/driver boards.

This SBC supports network and USB that will not be needed for the initial requirements, however it may be very useful when the student wants to connect his/her own peripherals later on.

The NC677 has an astonishingly low price compared against the other two SBC's with their given features. This board does not have an on-board battery charger or direct battery input therefore, if we use this in the final product then the charger module has to be built separately.

Table 21: On board features of the NC677

Feature	Description
CPU	VIA C3 733MHz MMX
Sound	VIA 1612 AC97
Display Card	AGP x4 Savage Twister T 32Mb shared memory
Network Interface	Realtek 8139C 10/100 Mbps
Storage	40Gb Hard disk 24x CDROM (5v)
Memory	256Mb (32Mb shared)
USB	Version 1.1 (12 Mbps)
Price	R 2,200.00

The hardware seems more than capable of handling the task at hand. The faster processor and improved bandwidth to the display card will prove

²⁹TFT (Thin Film Transistor) is a type of LCD which is more expensive while exhibiting very clear (or "crisp") and high quality graphics output with high resolution.

exceptionally useful when playing video files. A summary of the on-board features of the NC677 is shown in Table 21.

3.6.2. Windows

The same version of windows was installed on this board than the initial board. Therefore, we expected the same problems as with the first board because of Windows' native driver support.

The same tests that were described earlier were used to fill in the comments of Table 22.

MS Windows installed to the SBC with ease and very functional initial settings and configurations. The faster processor was noticed immediately in the basic working and many system restarts characteristic to a Windows system. It is indeed worth paying more for this machine when considering the speed at which it handled the tests compared to the previous two SBCs.

Table 22: Windows on NC677 before drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Low quality
Display Video	No
Sound output	No
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.6.2.2. Problems before additional software was installed

After a full installation of Windows, a few problems surfaced:

- 1 Video was unacceptable
 - 1.1 Resolution 640x480
 - 1.2 Video was very slow

1.3 Video was 16 colours

2 Sound did not work

3 Network did not work

1. Video:

The same problem with the video driver surfaced as with the previous two SBCs. The driver installed the same as before and this time the results turned out to be spectacular.

The resolution could now be set to a range of different resolutions. We chose 800x600, 16bit just to be consistent with the previous two. The display was fast enough to watch MPEG2 and DivX or MPEG4. These files seemed to play very smooth and with no errors or warnings. At this stage of the feasibility study the stronger processor and display card are showing their advantages.

2. Sound:

The sound driver provided by the manufacturers was installed. The driver detected the sound hardware and was fully functional.

3. Network

Network drivers were provided by the manufacturers. After the network was installed Windows only restarted once and the network was fully functional.

Windows' performance after the drivers were installed is shown in Table 23.

Table 23: Windows on the NC677 after drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	Yes
Sound output	Yes
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.6.3. Linux

Slackware Linux 10.0 was installed with no problems during the installation procedure. Everything was installed onto two partitions; one large partition mounted on root and one 512Mb swap partition.

The tests that were run with Linux on this SBC were the same tests described with the first SBC running Linux, these test results are summarised in Table 24. The requirements were all met, although there was one technical discrepancy during the tests. The technical discrepancy was that the frame rate tested with this hardware, measured with "glxgears", only produced 23 frames per second on a 800x600 16bit display configuration while we expected more from this Savage display hardware.

Table 24: Linux on NC677 before drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	Yes
Sound output	Yes
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

3.6.3.2. Problems before additional software was installed

After a full installation of Linux, one problem surfaced:

1. Video was acceptable (but not as expected)

1. Video

The hardware did play video but the CPU usage was very high. Both MPEG2 and MPEG4 videos played and seemed fully functional to the untrained human eye. The media player (Xine) gave errors that there were too many dropped frames and that the system was too slow to play any of these files. MPEG2/4 video played with 99% CPU usage. This was odd because Windows played the same video files with no errors.

This suggested that the CPU was not the bottleneck but rather the display card, its driver or the media player software. We know the display cards' frame rate is lower than expected. Therefore, we first try another display driver.

At the time of this writing, the Savage KM133 display card was not yet fully supported by stable³⁰ releases of Xorg. There exists a DRI (Direct Rendering Infrastructure) project on the Internet [26] that did support the Savage KM133 card in the form of regular snapshots of the drivers. DRI uses direct rendering to speed up the rendering process.

³⁰ Stable in this case meaning versions from which the source code does not change in time. Any further development will be made to the next version.

The following DRI drivers were downloaded from the Internet [26].

1. Common drivers
2. Savage drivers

The drivers did not install at first. The compiler complained about not detecting a configured kernel tree. A configured kernel tree is the Linux kernel source tree that had been configured previously with the "**make menuconfig & make dep**" commands in that order. The Linux kernel was configured with the previous ".**config**" file in the boot directory and "**make dep**" was issued to configure the kernel tree. The driver compilation again complained about not having a configured kernel tree. The kernel was recompiled, installed and the system restarted. After the new kernel was running the drivers compiled and copied into the X config directories.

X did not start up at all. Many errors occurred stating that unresolved calls were made by the drivers. These errors indicated that the drivers expected system calls that did not exist within the current Xorg (which was version 6.7.0).

Xorg 6.8.2 was downloaded and installed. The DRI drivers were installed again and compiled against the new 6.8.2 version. This time X started up with one error which stated that DRI was disabled due to insufficient permissions. The Xorg config file was edited and the following were added to the config file:

```
Section "DRI"
    mode 0666
EndSection
```

Section "DRI"

.....

.....

EndSection

The above states that a new subsection is to be created in the memory with some non-default values within this subsection [26]. Within this new section there are a few parameters that may be used with the same results. The first is to manipulate the DRI permissions in such a way that everyone has access to them.

Mode 0666

The above statement indicates that the default mode for DRI functions will now become "0666" which means that all users will have read and write access to the DRI drivers/features. The other method is to set up a certain group which will have access to the DRI functions.

Group "dri-group"

Mode 0660

The above statements indicate that everyone that is in the group "dri-group" will have access to the DRI drivers/functions. This method indicates the power and flexibility of Xorg, however it is not suitable for this project because everyone using this device must be able to watch video files and not only specified people in a certain group.

When X started up again, the drivers were fully functional and the speed according to "glxgears" indicated a difference of about 5x the speed of the hardware with the previous driver. Full screen frame rate measured with "glxgears" produced an average of 114 frames/second. Before the drivers were enabled "glxgears" produced 23 frames/second in full screen mode with the same resolution.

Video clips still proved to use 99% of the CPU.

Conclusion on video:

The problem was resolved, but will be given in the introduction of the next chapter because the solution involves optimization. After Xine was optimized everything worked fine and CPU usage was down to roughly 75% while playing MPEG4 video files.

Table 25: Linux NC677 after drivers

Specification	Comment
Use a CDROM disk	Yes
Display Text	Yes
Display Pictures/Graphs	Yes
Display Video	Yes
Sound output	Yes
Capability for making notes on this device	Yes
Some usability beyond the classroom	Yes

The problem lied with software and previous optimization in the pre-compiled packages. More proof that pre-compiled software is not the best route to follow in all cases.

The performance of Linux after the drivers were installed (shown in Table 25) indicated no difference to the required functionality. However the difference due to the speed of the new display driver was amazing. This is not yet optimization but rather correct installation.

3.6.4. Conclusion of NC677

At first sight Linux seemed to have better initial hardware support because Windows did not support various hardware whereas Linux did! After drivers were installed for Windows, everything worked fine.

Both Windows and Linux were fast enough to play MPEG2 videos as well as MPEG4 which has stronger compression.

For this processor board Linux is indeed cheaper than Windows.

This processor board can be used for the final product because it is fast enough and all features are working. If MPEG4 compression is to be used in the future then this processor board will produce the desired results because it can play MPEG4 video's with only 75% CPU usage. This SBC is also very small (100mm by 140mm) and can easily be mounted in a neat final product. This SBC is above the minimum requirements for the multimedia reader. It may be possible to use one of this SBCs variants with a slightly slower processor for the final product to lower the hardware cost.

3.7. Conclusion

Linux is indeed suitable for the task at hand because Linux could produce the same results as Windows with the same hardware. Linux proved to have hardware support for a large number of peripherals without the need for added device drivers. Device drivers for Linux do exist when new hardware are not supported by default.

Furthermore Linux can be optimized for better performance and smaller size. This feature will be explored in the next chapter.

Linux requires more skill and knowledge to install. This is not a problem for either the end user or the manufacturer because Linux is installed only once and can then be duplicated to create numerous exact systems. This will also be

true for upgrades, the upgraded system can at any time be copied to any target system. Thus eliminating the need to optimize each individual unit.

Table 26: Linux vs. Windows with respect to hardware support.

	Feature	Windows Before Downloads	Linux Before Downloads	Windows After Downloads	Linux After Downloads
NC546	Sound	No	No	No	No
	Video	No	Yes	Yes	Yes
	Network	No	Yes	Yes	Yes
ePanel	Sound	No	No	No	Yes
	Video	No	No	Yes	No
	Network	No	Yes	Yes	Yes
NC677	Sound	No	Yes	Yes	Yes
	Video	No	Yes ³¹	Yes	Yes
	Network	No	Yes	Yes	Yes

Note: The hardware is stated as supported only if the specific hardware is functional after driver installation, and not if the driver merely exists on the Internet.

Now that we are sure that Linux (which is free) can be implemented on the same hardware than Windows (which is not free) with almost the same results.

The summary given in Table 26 suggests that Linux is indeed better suited for this project than MS Windows.

³¹ The video was functional and quality was acceptable, but not as good as expected. The Savage4 has an AGP bus and therefore, 23 frames/second was quite embarrassing.

4. Chapter Four – Implementation

4.1. Introduction

In the previous chapter it was concluded that Linux is better suited for this project than Windows. It was also concluded that the NC677 board satisfies the requirements. Now we need to optimize Linux to be better suited for this project by speeding it up and making it smaller. This will only be done for the NC677 SBC because it is feasible to use the NC677 and not the other two for the reasons stated in the chapter on feasibility study.

In contrast to Windows, Linux can be optimized because we can choose how Linux works. We can choose how Linux is installed and we can choose how Linux looks. MS Windows does not offer any of those choices.

4.1.1. Fixing the CPU-usage problem

First we need to consider the advantage posed by the optimization of software. The problem from the previous chapter (Linux on the NC677) was that the video files played but with 99% CPU usage. We know that the processor on the NC677 is twice as fast as the other SBCs mentioned that could just about play the video files. Better CPU usage levels may be obtained with optimized software.

The problem could not lie with the video file because Windows could play the same file without any errors. The media player (Xine) could be the origin of the problem. An inspection of Xine showed that Xine-lib (the decoding libraries)

was compiled to use 3DNow which are special hardware functions contained in most i686-compatible processors. The VIA C3 processor does not contain the 3DNow functions and could therefore, not support binary executable files that use them. This processor has a SSE co-processor but not 3DNow functions. Xine-lib had to be recompiled from source code to enable SSE and disable 3DNow in the binary executables.

The optimization was initialized by setting three environment variables which is shown in Table 27:

Table 27: GNU Optimization variables for the VIA C3 processor

CHOST	"i686-pc-linux-gnu"
CFLAGS	"-march=i686 -msse -mmmx -mfpmath=sse -O2 -pipe -fomit-frame-pointer"
CXXFLAGS	"-march=i686 -msse -mmmx -mfpmath=sse -O2 -pipe -fomit-frame-pointer"

During the compilation of the source code, the optimization parameters shown in Table 27 was used and shown on the console for each file that was compiled. After compiling the source code Xine opened the video files and CPU usage was down to roughly 75% with no dropped frames reported. This was an improvement of more than 25% because Xine gave errors that frames were being dropped while the CPU usage was 99%. Therefore, SSE in stead of 3DNOW proved very effective.

Recompiling programs such as Xine to better suit the processor increases the interest to more complete optimizations. All packages that will be used can therefore, be recompiled for optimization purposes with better performance than standard i486. The use of i486 was for compatibility reasons, because all x86 processors (such as Pentium and AMD) are backward-compatible to the i486 architecture.

4.1.2. First remarks on optimization

The kernel of a system is provides the device drivers and some other services that links the hardware and the software together. Recompiling only the Linux kernel for a specific architecture provides any Linux user to speed up his/her OS considerably. This speed increase is due to the fact that the link between hardware and software is more effective.

MMX and other special features within a processor can be used to speed up performance. However these special features rarely gets used in a standard or non-optimized system. Enabling these special features on hardware that do not support them tends to choke performance (as we have seen earlier with Xine) therefore, they are not used in standard systems for compatibility reasons.

The GCC (GNU Compiler Collection) compiler supports various optimization switches such as MMX, SSE, 3DNow, Athlon, Pentium4 and many more. Therefore, if we can compile all the source code from scratch with the specific needed (and hardware supported) switches then the whole system can be optimized for the specific hardware. Then we are able to make the best use of the specific hardware at our disposal.

This chapter will consider some different approaches to optimizing the software, and some different distributions of Linux.

4.1.3. Choosing a distribution

Many Linux distributions exist in the world today and choosing one specific distribution to suit your own need is difficult and involves a great deal of thought and time. The features that attract interest to a specific distribution are the desirable features of an OS:

- Reliability
- Efficiency
- Maintainability
- Size

4.1.3.1. Reliability

We cannot guarantee reliability with the specific distribution, but we can guarantee that the newest source code does feature better reliability than older development sources. All source code has room for improvement, although mature code has had much improvement over its lifetime.

We may assume that all Linux sources are very reliable because of the fact that they are widely used for server systems [18].

4.1.3.2. Efficiency

All distributions are efficient. Efficiency for a specific architecture is inherent to each distribution, some distributions are optimized for predetermined hardware such as i486. This means that an example distribution which is precompiled for i486 will be efficient for a i486 processor but may not be compatible with an i386 processor. When an architecture is chosen such as i486 then it is probably chosen for compatibility reasons, because all processors are presumably backward compatible.

Some distributions are optimized during installation. Optimizing during installation does take longer than the more simple approach of just installing (copying the files to the target system). Once an OS is installed and optimized it is faster than a non-optimized OS, although optimization takes a long time. However an OS is installed and optimized once and used many times therefore, this approach has its immediate advantages.

4.1.3.3. Maintainability

The package manager is used to install and uninstall packages. Packages are software projects that are installed and used as add-on modules to the OS. This can be done with precompiled packages and source code.

Each distribution is configured in a special way making it unique. The differences between the configuration of some Linux distributions are sometimes as far apart as Linux differs from Windows. The administrator of the system will spend a lot of time configuring and maintaining the system therefore, he/she must be comfortable with the specific distributions' configuration methods.

4.1.3.4. Size

Size is very important to the outcome of this project. We need full functionality (as described by the requirements) on a small space because we are using a compact flash drive as the storage media. If a hard drive could be used then size would not be important any more. Packages can be uninstalled with the package manager to make a standard distribution smaller than usual while still achieving the required functionality. Or we may install the minimum system and add packages until every requirement is met.

The three different Linux distributions (or flavours) that will be explored in this chapter are:

1. Gentoo
2. LFS (Linux From Scratch)
3. Mini-Slack

4.2. Gentoo

Gentoo is a rising star of Linux distributions among Linux enthusiasts. Gentoo will suit a technically inclined person much better than a person who just wants to know the basics - "show me where to click for my e-mail". To install and maintain Gentoo effectively one has to care (to an unnatural extent) about what is going on inside your computers' internal working [18].

Performance and efficiency were the critical goals that led to the creation of Gentoo. The Portage software distribution management system is the key technology that separates Gentoo from other Linux distributions. Portage can

be used to build almost the entire Gentoo distribution from source code, and manage and upgrade that software as well [18].

Gentoo is sometimes referred to as the build-from-source Linux system. Most other Linux systems provide a set of pre-compiled packages to install and the user is never forced to build the whole system by himself. Gentoo provide pre-compiled packages as well as source code to enable the user to build the Linux kernel and all packages on the machine during installation [18].

The pre-compiled packages are pre-built with specific decisions already made for the user. By building the Linux system while it is being installed, the user can create a distribution that takes the following considerations into account [18]:

Considerations while installing Gentoo:

1. What processor is used
2. What hardware does the user want to support
3. What services does the user want to support
4. What software does the user want to install

4.2.1. What features are available

While the user compiles the packages he/she can choose which features of that specific package are included or excluded.

Gentoo enthusiasts like to configure, tune, tweak and update their Linux system continuously. The result being a computer system that runs faster and uses less system memory and disk space, than would be the case with standard off-the-shelf Linux distributions [18]. Installing Gentoo once and using it as a stable distribution is also possible.

4.2.2. Basic Gentoo installation process

In the Gentoo installation there are three stages. Each stage holds its own advantages and disadvantages and is described separately. Stage two is

basically a further developed version of stage one with some unavoidable choices already made by the developers. Stage three is a further developed stage two, also with some more unavoidable choices made by the developers.

4.2.2.1. Stage one

The least choices are made for the user and leaves a lot of configuring to the user to do. Building a complete system from stage one takes the longest time to complete.

4.2.2.2. Stage two

Building a complete system from stage two takes less time to complete than from stage one but is still not the fastest. More choices have been made for the user but there are still some tweaking to be done.

4.2.2.3. Stage three

Building a Linux system from stage three is the fastest but almost all of the choices and configuration are already made for the user by the developers.

4.2.3. Implementation of Gentoo

We did not know how much space was needed to install Gentoo therefore, it was installed on a hard disk first.

The steps taken to install Gentoo will be described in short.

4.2.3.1. Partition the disk

First the partitions were made with "cfdisk". Then the partitions were formatted with their own specific file systems.

Table 28: Gentoo file systems

Partition	Size	Mount point
/dev/hda1	100Mb	/boot
/dev/hda2	20Gb	/
/dev/hda3	20Gb	/LFS
/dev/hda4	1Gb	swap

All of the partitions shown in table 28 were not directly used for Gentoo. The partition "hda3" which is the third partition on the hard disk will later be used for the next system "Linux From Scratch" which needs another system to be built from.

After the partitions were formatted with their own native Linux file systems, they were mounted on their respective mount points under "/mnt/gentoo" of the local root system of the boot CD.

4.2.3.2. Extract stage one tarball

Stage one tarball was extracted to the newly created and mounted partitions. The stage one tarball provides the minimum set of tools to start compiling all other programs. After the stage one tarball is extracted some configuration files are modified to suit the specific computer.

4.2.3.3. Building the Kernel

The kernel will be built with the specific features of the specific hardware it will be used on. Building the kernel specifically for the hardware in question provides the computer with a high performance software core on which all software will run.

4.2.3.4. Installing system services

The system services runs in the background and makes the system more usable.

4.2.3.5. Installing all other software

All software will be installed from source code and therefore, this step takes a lot of time to complete. All dependencies will be taken care of during installation and therefore, pre-compiled dependencies will not surface later on. Installing from source code with special optimization switches ensures that the hardware will run at its full potential.

4.2.4. Gentoo – Concluding remarks

Gentoo produces a nicely optimized system that speeds up performance considerably beyond most pre-compiled versions. This SBC has a VIA C3 processor which is not standard therefore, there exists no pre-compiled versions that are optimized for this processor.

When Gentoo is installed from scratch and measures up to the requirements it takes up roughly 1Gb of disk space. This is not much considering that it is still **less than** 2.5%³² of most hard disks in use today.

Table 31 compares Gentoo with the other implemented distributions.

4.3. LFS (Linux From Scratch)

Linux from scratch lets the creator (the user) choose every aspect of the distribution. As the name implies, this is not a specific distribution but rather a complete build from scratch (source code). This has the same advantages that Gentoo exploits by compiling everything from source code. However Gentoo starts with a pre-built stage “tarball” that has some configuration settings already made. LFS systems can be made as small as 5Mb [25].

LFS has the initial problem with installation that one needs a compiler to start with. The compiler cannot be compiled first because we need a compiler to start with [25]. This circular dependency is tricky to implement in the real world.

³² 40Gb hard disks are becoming very rare. At the time of this writing 80Gb hard disks were standard.

4.3.1. LFS installation process

First we start with a host system which is another Linux distribution with specific features or requirements. The first requirement is that the host system has a compiler. If the host system does not have a compiler then we still need a compiler to start compiling Linux from scratch. The host also needs an available partition where the final LFS system will reside. This clean partition needs to be roughly 3Gb. The final LFS system will not take up all of that space, but the space is used during the compilation process. Other requirements are specific to the version of LFS; LFS is very sensitive to the kernel version and Glibc version. We have found that Gentoo provides the best host to start from, because Gentoo needs a strong compiler to be able to compile its own packages.

The process we followed for this project is as follows:

4.3.1.1. Build the toolchain

The toolchain consists of the following packages in the following order:

- Binutils
- GCC
- Linux-Libc-Headers
- Linux kernel header files
- Glibc

The toolchain is built in order to build the final and actual system. Some of the packages in the toolchain are used to resolve circular dependencies such as needing a compiler to compile a compiler.

All of the above packages are installed in a tools directory which is located on the new partition where the final LFS system will reside. A symbolic link from the host system `"/tools"` pointing to the tools directory on the target partition helps to ease the installation while automatically creating the needed hard links during the installation process.

4.3.1.2. Use the toolchain to reproduce itself

When the toolchain is built, there are a few considerations. What is the purpose of these new installed packages and why do we need to install them twice? The toolchain is built statically which means that the toolchain does not link to other installed programs on the host system. Therefore, creating the back door for separating the LFS system from the host system. If we want a truly independent LFS we have to link with packages that have no connection with the initial host system.

The following packages will be installed in the following order [25]:

Table 29: Linux from scratch core packages

1. TCL	8. Bzip2	15. Gettext	22. Bison
2. Expect	9. Gzip	16. Ncurses	23. Flex
3. DejaGNU	10. Diffutils	17. Patch	24. Util-Linux
4. GCC (pass 2)	11. Findutils	18. Tar	25. Perl
5. Binutils (pass 2)	12. Make	19. Texinfo	26. Udev
6. Gawk	13. Grep	20. Bash	
7. Coreutils	14. Sed	21. M4	

After Binutils is installed for the second time the tools directory is independent of the host system because the new packages in the tools directory only links with packages that were already statically installed. The toolchain is now ready to be "chrooted"³³ into in order to configure the rest of the LFS system.

4.3.1.3. Build the rest of the base system

After we have chrooted into the new partition where the LFS system will reside there are some settings to be made before any packages can be installed. The directory tree needs to be set up and some bootscripts need to be installed. There are also some configuration files to be created.

³³ Chroot (change root) is a method of booting into a system which is not necessarily bootable by itself but contains the necessary binary executables for a functional system.

After all the settings and configuration is done, the whole system is installed and made bootable.

4.3.1.4. Build the applications needed for this project

The LFS base system is now ready for use. The user can now choose which packages he/she wants and install them from source code. This enables the user to choose every aspect of the system.

Expanding the system the way the user wants, gives the user ultimate control over the system. Choosing the specific system software ensures that no unneeded or unwanted software will be installed or needed later on. The user may also choose which applications to install. If there are many different applications that provide one specific function, then the user may choose one or more of those he/she wants to use.

4.3.2. LFS – Concluding remarks

LFS is fast and small, but not small enough. When the LFS system is finished and measures up to the requirements, it uses around 1Gb of disk space. Cutting down on the size of the LFS system is quite difficult and very time consuming. All files must be deleted one by one, and you have to know what you are deleting. Deleting one single needed file can seriously damage any system.

Building a LFS system takes a long time and does push performance up to detectable levels beyond that of some precompiled distributions.

Table 31 compares LFS with the other implemented distributions.

4.4. Mini-Slack

Mini-Slack³⁴ is a smaller distribution of Slackware (which was used in the feasibility study) that seeks to present the user with one application for one purpose. Mini-Slack also presents the user with four basic functionalities such as web browsing, a sound player, a video player and development tools. Where the traditional Slackware would present the user with four or more web browsers Mini-Slack only has one web browser. Mini-Slack installs to a maximum size of 1.1Gb which is less than half the size of Slackware Linux.

Features of the standard Mini-Slack distribution include:

- Development tools
- Xine for watching Video's
- BMP for listening Sound files
- Firefox for Web Browsing
- Thunderbird Mail client

Mini-Slack comes with a package manager that enables the user to install and remove packages without the need of a compiler. Therefore, we do not need the compiling tools. Every package that is removed saves space and leaves more space for the user to utilize for his/her own files.

4.4.1. Installation of Mini-Slack

The setup scripts handles every step of the installation procedure and is almost fully automated. Linux installs fairly easy except for some issues that needs some thought from the user. These issues involve partitioning the system and, because we are not installing everything, which packages to install.

First of all we boot up with the CDROM disk containing Mini-Slack. After the CDROM boots up it is time to prepare the system by partitioning the drive. It is very important to know what we want to do with the system, because that would determine how the storage space will be partitioned. It was decided to

³⁴ Also known as "Zenwalk" at the time of this writing.

leave the system with one large partition because a lot of changes will be made to the system and storage space is quite tight. It is possible to transfer everything that is already configured to a backup CD/DVD and then duplicate it onto other storage media from there.

The original specifications does not state the disk space requirements. Therefore, there are no guidelines for partitioning the storage media. We want to use a compact flash disk for storage. Linux must then be fitted on a standard³⁵ disk size with enough room to spare for user files. Compact Flash drives with less storage space will cost less than those with more storage space.

We are using Compact Flash to store the OS therefore, we cannot dedicate a SWAP space, because each byte on a flash drive can only be written to a certain amount of times (roughly 10,000 cycles) and SWAP will use up those cycles very quickly.

We have very little space that is to be mounted on "/" (root). Installing everything and removing packages one by one will not be a good approach because some packages may be compiled to use other optional packages and bugs (in the form of broken links) may then surface when the final product is already in use. Thorough testing after each package is removed, is not possible because that approach would take too much time. We suggest starting with a "minimal³⁶" system (with development tools) and installing packages that are needed one by one until the whole system covers the requirements.

Mini-Slack uses mainly precompiled packages. We want to install packages from source code for the following reasons. Installing from source code has the advantage that packages will be optimized for the given processor. Installing directly from source will also have the advantage that dependencies will be taken care of as we go along. Most packages will not configure or compile if all required dependencies are not met. Packages will give warnings if optional

35 Standard Compact Flash disk sizes are 128Mb, 256Mb, 512Mb, 1024Mb, 2048Mb, 4096Mb and 8192Mb

36 Minimal meaning the smallest number of packages that can be installed while still creating an expandable system.

dependencies are not found during the configuration script and will result in not linking with those packages (which is exactly what we want). In contrast to precompiled packages that may already be linked to some optional packages that we do not need, in such cases the packages needs to be recompiled to correct that problem. Installing from source with a distribution that enables the user to uninstall precompiled packages has the advantage that the development tools may be removed when all the compiling is done, freeing up more space.

We start with a clean installation with the following packages:

Table 30: Our base Linux system

aaa_base	diffutils	hotplug	reiserfsprogs	utempter
aaa_elflibs	e2fsprogs	less	sed	util-linux
alsa_lib	etc	libusb	shadow	X11-base
alsa_oss	findutils	lilo	Startup-notification	X11-font-misc
alsa_utils	gettext	mod-int-tools	syslinux	X11-font-scale
bash	gettext_tools	ncurses	sysvinit	bzip2
bin	Glibc-solibs	openssl-solibs	t1lib	gzip
coreutils	Glibc-zoneinfo	pkgtools	tcsh	tar
cxxlibs	gawk	procps	udev	zlib
devs	grep	readline	usbutils	

Size: 215Mb

Then we added development packages:

- gcc
- gcc-g++
- Glibc-2.3.5
- X11-devel

Total size: 366Mb

4.4.1.2. Compile the kernel

The procedure followed is given in the appendix. The reasons for compiling the Linux kernel is simple, firstly a specifically compiled kernel is faster and uses less memory. The first step into transforming Mini-Slack into the system we want for this project – an optimized core. Secondly we need a configured kernel tree to enable the Savage4 drivers to be compiled.

4.4.1.3. Compiling the Savage4 drivers

Secondly we install the savage drivers to see if everything is in working condition before doing much more work.

Installing the display card drivers as done in the previous chapter on the feasibility study speeds up the graphical rendering speed.

After the compilation process the configuration files for Xorg must be edited to ensure that the correct drivers will be used together with the needed selected options. Then tests were conducted as shown in the appendix and indicated that the rendering speed was 114 frames per second on 800x600 16 bit resolution.

4.4.1.4. Compiling user applications from source

After the display drivers were successfully installed all user applications were compiled from source code. All dependencies were installed prior to installation of the specific user applications.

Because most user applications use the same dependencies (such as GTK+-2 and Glib-2) they were installed to satisfy more than one user applications' dependencies.

4.4.2. Mini-Slack – Concluding remarks

Two approaches were taken:

1. Start with a new installation of minimal components and install precompiled packages until the requirements were met and
2. Start with a new installation of minimal components and install packages from source until the requirements were met.

Both proved to work, however there were some irritating dependencies in the pre-compiled packages and all of the dependencies had to be installed before the programs worked. The dependencies did not make sense, for example AbiWord did not run before ESD was installed and ESD is gnome's sound manager. ESD is obviously not needed to write a simple text file and takes up valuable space. This approach produced a working system within 480Mb leaving 32Mb for user files when we use a 512Mb CF2 card.

Starting with a "minimal" system and compiling everything from source took a long time but proved to be worth the effort. All packages ran faster than the pre-compiled packages and annoying optional dependencies were eliminated because each package was compiled to link only with the other packages already compiled/installed. This approach ended up as 372Mb after the development tools were removed leaving 140Mb for user files when a 512Mb CF2 is used. The final optimized Mini-Slacks' system performance did not surpass that of Gentoo or LFS but the size was considerably smaller than that of Gentoo and LFS. Mini-Slack performed all the needed functions while still fitting on a 512Mb CF2 card.

4.5. Conclusion

Table 31: Comparison of the optimized distributions

Name	Time taken to install (733MHz)	Relative Performance	Size
Gentoo	1 Week	Very fast Equivalent to LFS	~ 800Gb
LFS	3 Weeks	The fastest (Roughly twice the speed of Windows)	~ 800Gb
Mini-Slack (precompiled packages)	5 Hours	Acceptable (As fast as Windows)	~ 480Mb
Mini-Slack (expanded with source code)	2 Days	~25% slower than Gentoo and LFS	~ 370Mb

As seen in Table 31, Gentoo and LFS take a huge amount of time and the results are spectacular. Unfortunately the main goal is not only speed optimization but also small size. Slackware Linux produced a working system which is both fast and small enough to fit on a 512Mb flashcard with room to spare for user files.

With enough knowledge and time the LFS system may prove to be smaller than the system built from Slackware Linux. However the whole system can be configured much easier with Slackware, and non-technical people will work with this media reader later on.

When compared with the speed of MS Windows, the pre-built Mini-Slack demonstrates almost the same speed. The optimized Mini-Slack runs at roughly 150% the speed of Windows. Where LFS and Gentoo Linux roughly doubles the speed at which Windows performs on the exact same hardware.

The system is now ready for testing “outside the laboratory”. This is a real world project and must be tested in the real world. Features such as user friendliness, stability with various users and users' impression can only be tested where it will work in the future.

5. Chapter Five – Evaluation

5.1. Introduction

This is a real world project therefore, it must perform acceptable in real world situations. Due to the fact that the system is built up from various subsystems, each subsystems' working may be changed in any way if the interfaces with other subsystems stay the same. Then it is possible to fix various problems in different subsystems without affecting other working parts of the system.

5.2. How good is the design?

In any system there is room for improvement. This system consists of two subsystems namely hardware and software.

5.2.1. Hardware

The hardware is commercial and tested extensively therefore, hardware functionality might not be a problem. The problems that may be encountered with the hardware are packaging, heat, power usage, battery life, weight and future upgrades. We cannot do much about future upgrades, weight or packaging. We can do something about power usage, which in turn affects heat and battery life, by choosing low power hardware.

5.2.1.1. Packaging

The packaging of this hardware poses the problem that the hardware may not be sufficiently ventilated causing it to overheat. Correct packaging, which advances ventilation at the needed locations, on the other hand solves the heat

problem by extracting the generated heat sufficiently. This will be left to the packaging experts.

Packaging can also make a product attractive or not. The concept of “eye candy” must be used for a product to be more successful. If a good product such as a cellphone does not look attractive enough, it will not be as successful as the other phones on the market.

5.2.1.2. Heat

Heat is generated by the hardware's normal operation and cannot be used in a closed area where the heat builds up. The generated heat must be taken away from the hardware to keep the whole system from overheating.

5.2.1.3. Power Consumption

This multimedia reader must use very little power. Lower power usage in turn means longer battery life and less heat generation. A decision to use the compact flashcard as storage media lowers the power usage more with the harsh penalty to available storage space for both the OS and the user. However storage space is not as important as low power consumption in this project.

Table 32: The product's total power consumption.

Component	Current usage	Comments
SBC	1000mA	Main processor and peripherals.
Screen	800mA	10.4" TFT LCD Screen.
CF2	50mA	512Mb storage media.
CDROM	100mA	High power usage when in use.
Mouse	~0mA	Not practically measurable
Keyboard	~0mA	Not practically measurable
Total	~2000mA	An estimate of maximum power usage.

5.2.1.4. Battery life

Batteries are rated in mAh, which indicates the current that the battery can deliver for one hour. Example if a battery is rated 4500mAh then it can produce 4500mA for one hour or 450mA for ten hours. In our case this battery can give 2000mA for a little more than two hours.

5.2.1.5. Weight

Nobody wants to carry a heavy piece of equipment around therefore, the weight is an important factor to the success of this project. Very little hardware is used and cutting down on weight is very difficult. The net mass of each individual piece of hardware is given in table 29. The keyboard and mouse is externally connected and not considered as part of one standard package.

Table 33: The product's estimated total weight.

Component	Weight	Comments
SBC	350g	The data sheets says 340g.
Screen	330g	Including inverter board and connection cables.
CDROM	210g	5v Laptop CDROM
CF2	5g	Low weight, low power.
Battery	300g	Standard laptop battery
Keyboard	N/A	Externally connected
Mouse	N/A	Externally connected
Packaging	≤305g	By estimation.
Total	1.5kg	Same as a single first-year physics textbook.

5.2.1.6. Hardware Plug-ins

Upgrading this multimedia reader in the future is possible, because there are unused I/O ports that enables the user to connect arbitrary hardware to this device. Therefore, it will not be necessary to redo the whole project just because the users want to add some random hardware (such as a printer to

print some notes) to this multimedia reader. Most add-on upgrades are available as USB devices.

This multimedia reader supports all USB devices from companies who make their drivers available to Linux.

5.2.1.7. Price

This multimedia reader must be low-cost. Much was done to lower the cost in this project while still achieving minimum functionality. Minimum functionality was set quite high due to video output. Playing video files is a tough constraint on hardware because it implicates video processing which can only be done in real-time on faster processors.

Table 34: The product's estimated total cost of quantities of 1000 units

Component	Cost	Comments
SBC	R1400	The core of the hardware
Screen	R1500	Including inverter board and connection cables.
CDROM	R100	5v Laptop CDROM
512Mb CF2	R400	Low weight, low power.
256Mb RAM	R250	More RAM would be advisable if we use more professional Office tools such as KWord or OpenOffice.org
Keyboard	R50	Approximate cost.
Mouse	R50	Approximate cost.
Packaging	N/A	Not available at the time of this writing.
Total	≤R4000	Little less than your text-books for one year.

This project does exceed the budget by some 30%, yet mass production has proven itself to lower costs in the past. Even though this multimedia reader costs too much, it definitely costs less in the long run compared to textbooks.

Costs can be cut further by using other technology for the screen such as the \$100 PC described in the chapter six. Time may also cut the cost, taking into account that the current screen-technology's price came down by roughly 50% during the course of this project.

5.2.2. Software

The software is optimized to use two special features:

- MMX
- SSE

These features make it possible to use today's graphics on yesterday's processor speeds. The software is compiled to use the minimum set of dependencies.

The software is also chosen to present the user with a comprehensible system within the minimum disk space requirement. The final system uses ~370Mb of disk space presenting the user with roughly 140Mb of space for user files if a 512Mb CF2 is used.

The final system also has the option to expand with software compiled for this optimized system. Unfortunately this option poses the problem that the user must know the root password which can be dangerous in the user's hands.

5.3. *How good does this system perform?*

The system is not fully optimized from scratch because that was not necessary for this project. The optimizations that were done in chapter four reached results that were more than good enough to satisfy the requirements.

The system is a true multi user and multitasking machine with surprising capabilities. The details of a few aspects are given in table 31.

Table 35: Benchmarking.

Test	Performance	Comments
"glxgears" 800*600 16 bit colour resolution	114 frames/second	Displaying video uses 25-29 frames per second
"glxgears" 800*600 24 bit colour resolution	39 frames/second	Displaying video uses 25-29 frames per second
MPlayer (mpeg2)	51% CPU usage (average)	Playing video with light compression
MPlayer (mpeg4/DivX)	78% CPU usage (average)	Playing video with heavy compression
XMMS (OGG/mp3)	30% CPU usage (average)	Playing compressed sound with heavy compression
Making notes	1% CPU usage (average)	This test shows the low overhead for the window manager and other background services
Saving notes	User space for roughly 150 000 A4 size pages of text	The notes made in accordance with the study material will be enough for most students.

Using benchmarks to measure system performance does not mean much to anyone who is not technically orientated. Knowing that your display card can render 114 frames per second does not mean much, especially if you take into account that video files require roughly 25 to 29 frames per second. The given information only tells us that the display-hardware is more than fast enough to watch video. The frame rate cannot be directly linked to the processing power. How do we measure if there is enough processing power available to play certain types of files with their various compression techniques with a benchmark? Measuring the processing power does not indicate a systems ability to integrate certain tasks. Some integrated tasks are shown in Table 35.

This system will not necessarily be used as a computer, even though it has that capability. Therefore, running benchmarks that test the computational power of this multimedia reader does not supply useful results for this specific project.

We must devise a test plan to make sure if the multimedia reader reaches its goal which is to present the student with multimedia enriched study material. The results that come from such a test cannot measure the efficiency of the multimedia reader in the short term because if the problem lies with the student's motivation to learn then we cannot blame the study material or its medium. However we may in the long term be able to determine whether this multimedia reader is better than the current method or not.

5.4. How can this system be used in its final environment?

This multimedia reader is developed to present the student with enriched study material. The user can do four tasks on this machine namely read study material, watch video's from study material, listen to sounds about the material, make notes in accordance to the study material. Sound, video and text together form the enriched study material.

5.4.1. CDROM

The multimedia reader uses a CDROM as mass data input for all the study material that will be shown to the user. The CDROM has a large capacity of about 700Mb, which was standard at the time of this writing. Distributing CDROM disks to all the students costs roughly R1-60 a piece, at the time of this writing plus labelling and postage and packaging. Therefore, this is a very economical solution. The CDROM will then be used to store all the HTML files, picture files, sound files and video files.

The CDROM can be used to upgrade the multimedia reader at a later stage.

5.4.2. Mouse: “Core pointer device”

The mouse will be used to launch applications and for browsing through the study material.

5.4.3. Keyboard: “Core character input device”

The keyboard will be used to create and modify notes made in accordance with the study material.

5.4.4. Storage media

The storage media will primarily hold the OS and other installed applications while the free space can be used to store any arbitrary data from the user.

The storage media was not initially meant for the user, for that reason this project uses a Compact Flash module in stead of a hard disk used in normal computers and laptops. Furthermore there is not enough space to store study material by itself but rather the notes made in accordance with the study material. If this notes only consist of text, then the available storage space³⁷ will be more than any user could possibly fill alone.

5.4.5. Processor

The processor used in this project is fast enough to do common everyday calculations in acceptable times without being optimised. This processor also exhibits special features not found in standard architectures such as i486 or i686. Special features such as SSE and MMX (when they are used) speeds up the performance considerably. SSE is special functions and registers that were added to the CPU to enhance performance in several ways [49]. This enables the processor to work with larger numbers and floating point values that takes considerable amounts of time on standard architectures such as i486. MMX (Multimedia Extension) are specialized features dedicated to multimedia and can speed multimedia processing up by 300 to 400% [28].

³⁷ Storage space practically achieved was not less than 130Mb.

If we compile the OS (Linux) in the correct manner then all of these special features gets used effectively to speed up overall system performance. The specifications of this reader do not call for a strong computational device, however the user may soon want to use his reader for computing purposes later.

5.4.6. Screen

The screen resolution is high enough to read large chunks of text from. This is useful because a lot of students will be reading all of their information from this device.

5.5. People's views of this system

The survey that was done asked people three questions:

What makes this multimedia reader better than standard computers?

This question allows people to comment on their expectations from this device, whether they know about Linux or not.

What makes other available hardware better than this device?

This question forces people to think about technology and the usefulness of other hardware. This product has room for improvement, this question forces some comments that may improve on the system.

Will you personally like to learn from this device?

This question's answer will depict if the system reaches its goal or not. If students do not want to learn from this device, then we have failed.

This survey was not intended to be a complete marketing study, nor a statistical study on its own. This questionnaire simply tried to establish whether or not people would like to use a device such as this.

From each category there were 5 people who willingly shared their time to assist in assessing this project.

5.5.1. People with a strong computer background

Most people would like to have one of his/her own.

Good display quality.

The speed is faster than expected from the hardware.

One person prefers hard copies of study material.

Most people would like to learn from this device.

5.5.2. People without a computer background

All people would like one of his/her own.

The graphics are better than expected.

More comprehensible than conventional computers.

Most people would like to learn from such a device.

5.6. Conclusion

This device has a certain attraction to it. Most people have the direct comment that they too would like to own one of these devices. Not all people would like to study from such a device, which is understandable if you prefer hard copies to soft copies.

A complete manual will have to accompany this device which covers everything from the basics up to advanced settings.

A more complete study must be done to determine what optional programs must be used with this device. We cannot have a pre-compiled version of every single program that each user would like to install later on. On the other hand there is not enough space to equip each one of these devices with a compiler for the user to compile his/her own programs.

6. Chapter Six – Conclusion

6.1. Introduction

The combination of the NC677 SBC and Linux OS which is optimized for MMX and SSE satisfies both cost and functionality requirements. With this combination it is possible to use a 512Mb Compact Flash II (CF2) storage card. The CF2 card stores both the OS and the user's data. The CF2 storage card uses much less power and does not have moving parts. Moving parts implicates normal wear-and-tear in everyday use. Additional USB storage can be used by any user to expand the data storage capacity. Additional applications may be installed by the user for own use, creating a custom environment with which the user may feel more comfortable with. Expansion cards for the PC104 bus do include battery packs and charger modules which makes this system portable. Portability is stated in the requirements. Developing our own battery packs will definitely lower the total cost of the product together with mass production.

This project has delivered some interesting results with open source as the basis. The only cost linked to the system is the cost of the hardware, because all software used are free of charge and royalties.

We now have a complete system within the price range working at an acceptable speed and presenting its owner/user with the following features:

- CDROM³⁸ as primary³⁹ input media
- High quality, low power TFT LCD screen
- Web browsing (for reading study guides)
- Playing music⁴⁰ files
- Playing video⁴¹ files
- Ability to store⁴² additional notes
- Ability to install additional user applications⁴³

6.2. Comparison with recent products

Some new products surfaced near the end of this project and is therefore, not discussed in the literature study. One example is "The Pandora PC" which made it to the market in September 2005 [52], another is the "\$100 PC" which will be completed within two years of this writing [53].

These products are aimed at producing an affordable Personal Computer for the general public. This project, in contrast to the above mentioned, aimed to produce a computer that can do one specific task which is "to present a learner with enriched study material".

6.2.1. Comparison with the MAC-mini

The MAC-mini is a very modern tool, though it is too expensive to be used in this project. The MAC-mini costs roughly R3500 with no screen, no mouse or keyboard [51]. Adding the screen and other peripherals will cost extra.

The MAC-mini has faster hardware than the hardware used in this project. The hardware's speed does not create a problem for what the project demands

38The CDROM is the low-cost, high-capacity data storage and transfer media.

39Network, USB or an external modem can be used later for additional input/output channels.

40 File formats include OGG, FLAC, wave and CD audio which are all non-patented formats.

41 MPEG2 and MPEG4 can be played, again all non-patented formats are supported.

42 Storage space equivalent to a 128Mb USB thumb drive is available to the user for storing any arbitrary data.

43 Added user applications will share the same storage space as the user's data.

therefore, the speed is not a problem. Both the MAC-mini and the Wintel-mini costs a bit less than this project, but does not include the basic peripherals such as screen, keyboard and mouse.

Table 36: Comparison against our specifications

Specification	MAC-mini/Wintel-mini	Multimedia Reader
Use a CDROM disk	CD + DVD	CD
Screen	Not Supplied	10.4" colour TFT LCD
Display Text	Yes	Yes
Display Pictures/Graphs	Yes	Yes
Display Video	Yes (4x AGP)	Yes (4x AGP)
Sound output	Headphone jack	Headphone jack
Capability for making notes on this device	40Gb storage	140Mb
Some usability beyond the classroom	Yes	Yes
Price	R3500 + Screen and other peripherals	< R4000

6.2.2. Comparison with the Wintel-mini

The Wintel-mini or rather "Pandora PC" is a clone of the MAC-mini with almost the same specifications and price [52]. The comparison between the two products can be seen in table 36. The specifications we are looking at will be exactly the same between the Pandora PC and the MAC-mini.

6.2.3. Comparison with the \$100 PC

The R670 (\$100) PC is a strip down version of a PC aimed at advancing education. The machine will be ready within two years of this writing, but will not be sold in the United States [53].

The comparison with the \$100 PC and this project is shown in Table 37. All the specifications for the \$100 PC was not set out at the time of this writing.

Table 37: Comparison against our specifications

Specification	\$100 PC	Multimedia Reader
Use a CDRom disk	No	CD
Screen	11.8" colour rear projection Screen	10.4" colour TFT LCD
Display Text	Yes	Yes
Display Pictures/Graphs	Yes	Yes
Display Video	Not Available	Yes (4x AGP)
Sound output	Not Available	Headphone jack
Capability for making notes on this device	Not Available	140Mb
Some usability beyond the classroom	Yes	Yes
Price	~R700	< R4000

If we were able to use the screen that MIT uses for the \$100 PC, then we could perform better with respect to price. The screen we use at the moment is a high-quality screen and costs roughly twice the \$100 PC's price.

Further price cuts may be brought in by developing our own processor board. This may cut the cost of the board by selecting only needed hardware peripherals such as disk controllers and screen driver hardware. Then we can choose not to include unneeded hardware such as network connectivity and infra-red (IRDA).

6.3. Conclusion

This project focused more on the software implementation than the hardware. How well is the software suited for its application in the multimedia reader? Most computer generated graphics (such as animation movies which is multimedia) are done on Linux [18]. But does that necessarily mean it will produce better results out in the field where it will ultimately be used? We cannot predict the future or if most of the users will find either of the two OS's

more accommodating. There will surely be some people who will be willing to pay the R1300 extra to have Windows run on this device rather than Linux. We chose Linux to lower the total cost and received more than we bargained for.

6.3.1. Power supplies

The Wintel-mini and the MAC-mini are portable devices, however they cannot be used without an external power source. This project was developed using an external power source mostly because a power supply was not defined as part of this project, but still has the option to use only batteries or other low power alternatives. All hardware was chosen to minimize power consumption, and therefore the option to use alternative power sources remain open. The \$100 PC does not clearly state its power source and therefore we can not yet compare it to this product with respect to power source.

6.3.2. Price considerations

Price plays a significant roll in most products today. We have made several design decisions to ensure that this product is as low cost as possible. However, if it costs too much then very little will sell - ultimately causing the project to fail.

In our attempts to cut the price some hardware were discarded because it will not be used by most students. This did not have such a huge effect on the total cost of the system. When Windows was replaced by a free OS, the cost dropped by almost 20%. This suddenly made the system as a whole much more attractive.

Primarily we used Linux because it is free for everyone to use and not because it has strong multimedia capabilities. Should we have used Windows to be better suited for the environment where this multimedia reader will be used? There may be more reasons why 90% of the desktop market use MS Windows than Microsoft's effective marketing. Maybe Windows is better suited for people in rural areas because it may be a more comprehensible system, and

maybe Windows does not “break” that easily. The questionnaire described in chapter five indicated people's affinity to high quality graphics and fast speed. Linux and Xorg satisfies that need better than MS Windows according to some additional comments also stated in the comments. By setting the correct user rights, Linux exhibits strong protection to itself by not allowing normal users to alter and ultimately damage the system.

6.3.3. Which is better: Windows or Linux?

A recent study conducted at the North West University indicated that half of the employees at the university do not know what Linux is. The question was presented as a multiple choice with four possible answers, these results are shown in table 38.

Table 38: Which is better: Windows or Linux?

Option	Result
What is Linux?	49%
Both are equal	10%
Windows is better	19%
Linux is better	22%

These are educated people who voted, and still they do not know about Linux. If we ignore the people who do not know what Linux is, then Linux is slightly more preferred according to the opinion poll. However taking only the opinion of higher educated people who have heard of Linux does not necessarily mean that Linux will be better suited for students in rural areas.

Only time will tell if Linux is usable for the task at hand. Extensive testing can never compare to sending the product out into the field where it will ultimately be used.

7. Bibliography

- [1] <http://www.iuphar.org/sections/teaching/docs/PRES1.doc>
Downloaded on March 5, 2004
- [2] <http://www.e-learningzone.co.uk/index.html>
Downloaded on March 5, 2004
- [3] "Sustainable Internet access for the rural poor? Elements of an emerging Indian model". J. James; Development Economics; Tilburg University
- [4] "SOME REFLECTIONS ON THE DIGITAL DIVIDE BASED ON A BROADENED VISION OF TECHNOLOGY", ASJ Helberg, R Goede and S Strijbos. Proceedings of the 9th Annual CPTS Working Conference (2003)
- [5] "Breaking the digital divide: Implications for developing countries", Elena Murelli, Commonwealth Secretariat and SFI Publishing 2002.
- [6] Intelligent Assessment of distance learning; Flora Chia-I Chang; Elsevier Science Inc, 2002
- [7] Learner outcomes in a asynchronous distance education environment; Alan D. Carswell; Academic Press, Inc, 2002
- [8] Students' experience with collaborative learning in asynchronous Computer-Supported Collaborative Learning environments; Silvia Dewiyanti, Saskia Brand-Gruwel, Wim Jochems, Nick J. Broers; Computers in Human Behavior 2004

- [9] "QUESTIONS ON THE DEPLOYMENT OF A DIGITAL MULTIMEDIA READER", ASJ Helberg AND AC Janse van Rensburg
- [10] "EBooks: Evolution, Not Revolution, in Book Publishing", Reprinted by permission of Seattle Book Company from The Rosetta Bulletin posted February 9, 2004, Downloaded from:
[http://12.108.175.91/ebookweb/stories/storyReader\\$2549](http://12.108.175.91/ebookweb/stories/storyReader$2549) on March 11 2004
- [11] http://manybooks.net/eBook_readers.php
- [12] <http://www.gemstar-ebook.com/ebcontent/devices/default.asp>
- [13] <http://www.techdigital.co.za/category.aspx?categoryID=179&affid=33>
- [14] Fundamentals of operating systems; A.M. Lister; Macmillan, 1975
- [15] Operating Systems; H. Lorin, H.M. Deitel; Addison-Wesley, 1981
- [16] Operating System Concepts - Third edition; A. Silberschatz, J. Peterson, P.Galvin; Addison-Wesley, 1991
- [17] The X Window System – A user's guide; Niall Mansfield; Addison-Wesley Pub. Co., 1991
- [18] Linux Bible – 2005 edition; Cristopher Negus; John Wiley & Sons, 2005
- [19] Fluxbox Homepage; downloaded from <http://fluxbox.sourceforge.net/> on 25/04/2005
- [20] Teach yourself DOS + Virus Alert; F. Pieterse; R.J. Pieterse, 1993
- [21] Operating System Elements – A user perspective; Peter Calingaert.
- [22] Ten O'Clock Tech – Better Browser Now The Best
[Arik Hesseldahl](#), 09.29.04, 10:00 AM ET downloaded from
http://www.forbes.com/2004/09/29/cx_ah_0929tentech.html?partner=tentech_newsletter
- [23] **AbiWord vs. MS Word**
Posted by Team Flexbeta on 01 September 2004 (120073 views) Rating: 4.19. Downloaded from:

<http://www.flexbeta.net/main/articles.php?action=show&id=78>

- [24] Xmms's home page; downloaded from: <http://www.xmms.org/>
- [25] Linux From Scratch - Version 6.0; Gerhard Beekman; Iuniverse Inc, 2000
- [26] <http://dri.freedesktop.org/snapshots/>
- [27] <http://www.slackware.com/>
- [28] xiph.org: the Ogg project downloaded from: <http://www.xiph.org/ogg/>
- [29] mp3licensing.com downloaded from: <http://www.mp3licensing.com/>
- [30] 'Why Tastytronic Records prefers ogg' downloaded from:
<http://records.tastytronic.net/ogg.html>
- [31] 'Ogg Vorbis RC2 Released' downloaded from:
<http://www.mp3newswire.net/stories/2001/oggvorbis.html>
- [32] What is Ogg Vorbis? Downloaded from: <http://www.vorbis.com/intro.html>
- [33] 'Theora FAQ' downloaded from: <http://www.theora.org/theorafaq.html>
- [34] <http://www.x.org>
- [35] Beyond Linux from scratch downloaded from:
<http://www.linuxfromscratch.org/blfs/view/stable/index.html>
- [36] Windows discussion forum downloaded from:
<http://www.annoyances.org/exec/forum/win98/t1057252212>
- [37] <http://www.xfce.org/>
- [38] <http://www.kde.org/>
- [39] <http://www.kde-apps.org/>
- [40] <http://www.sosdg.org/~larne/w/Introduction>
- [41] <http://developer.kde.org/~wheeler/juk.html>
- [42] <http://www.mozilla.org/>
- [43] <http://browser.netscape.com/ns8/>

- [44] <http://xinehq.de/index.php/home/>
- [45] <http://www.mplayerhq.hu/>
- [46] <http://kate.kde.org/info.php>
- [47] <http://www.openoffice.org/>
- [48] <http://www.microsoft.com/>
- [49] <http://www.tommeseani.com/SSE.html>
- [50] <http://www.apple.com/macmini/specs.html>
- [51] Mac Mini: The Emperor's New Computer by Jorge Lopez, MCSE;
Technology Insider. Downloaded from:
<http://www.divisiontwo.com/articles/MacMini2.html>
- [52] <http://www.legitreviews.com/article.php?aid=249>
- [53] Closing the technology gap, Popular Mechanics – August 2005.

8. Appendix A – Compiling the kernel

First we recompile the kernel. Recompiling the kernel speeds up the system considerably, this will come in handy considering that it takes a lot of processing power to compile packages from source and obviously a lot more time on a slower computer. This is not the only reason, compiling the savage drivers for the video card needs a configured kernel tree and one way to obtain one is to build it yourself. Running “make dep” on top of the kernel tree does not work under all circumstances.

We downloaded the 2.6.10 kernel and proceeded with the following:

```
root[/usr/src/] tar -jxf /home/erich/linux-2.6.10.tar.bz2
root[/usr/src/] cd linux-2.6.10
root[/usr/src/linux-2.6.10] make menuconfig
root[/usr/src/linux-2.6.10] make all
root[/usr/src/linux-2.6.10] make modules
root[/usr/src/linux-2.6.10] make modules_install
root[/usr/src/linux-2.6.10] cp System.map /boot/System.map-via-c3
root[/usr/src/linux-2.6.10] cp arch/i386/boot/bzImage /boot/kernel-via-c3
```

add the following lines to “/etc/lilo.conf”:

```
image = /boot/kernel-via-c3
  root = /dev/hdb1
  label = Slack-via-c3
  read-only
```

Run the boot record program with the command: “lilo”.

reboot the system with the command: “reboot”.

9. Appendix B – Installing the Savage4 drivers

We have two tared archives:

1. common-20050618-linux.i386.tar.bz2
2. savage-20050618-linux.i386.tar.bz2

Untar both archives:

```
erich[/home/erich/] tar -jxf common-20050618-linux.i386.tar.bz2
erich[/home/erich/] mv dripkg dri-comm
erich[/home/erich/] tar -jxf savage-20050618-linux.i386.tar.bz2
```

As root:

```
erich[/home/erich/] su
"Give root password"
root[/home/erich] cd dri-comm
root[/home/erich/dri-comm] ./install.sh
root[/home/erich] cd ../dripkg
root[/home/erich/dripkg] ./install.sh
root[/home/erich/dripkg] exit
```

To test the drivers make sure the `"/etc/X11/xorg.conf"` file has the following lines:

In the "Module" section:

```
Load "glx"
Load "dri"
```

In the "Device" section:

driver "savage"

At the end of the file:

Section "DRI"

mode 0666

EndSection

Make sure the "~/.xinitrc" reads:

exec xterm

Then start up the X server with the command:

startx

A very simple and very basic window manager known as TWM will start up and display a basic virtual console known as "xterm". Use that console to test the video card by typing: "glxinfo | grep render". This function will reveal if direct rendering is enabled or disabled together with the renderer which would be Savage Twister. If that works it would look as follows:

direct rendering: Yes

To test the speed of the display type "glxgears". This command draws rotating gears on the screen as fast as the display card can handle while calculating the speed every five seconds. Press `ctl+c` to stop glxgears.

The savage card was tested and produced: 464.2 frames/second.

To exit that window manager type "exit" and press enter.

10. Appendix C – Compiling programs on Mini-Slack

10.1. XFCE

Depends on:

- 1 Gtk+-2.6.4
 - 1.1 libtiff
 - 1.1.1 libjpeg-6b
 - 1.2 Xorg
 - 1.2.1 libpng
 - 1.2.2 fontconfig
 - 1.2.2.1 expat
 - 1.2.2.2 freetype
- 2 Pango
 - 2.1 Glib-2.6.3
- 3 ATK
 - 3.1 Glib-2.6.3
- 4 libXml2
- 5 libdbh

The minimum list of dependencies was installed to use the XFCE window manager. The packages installed with no problems and will not be discussed.

10.2. Mozilla-Firefox

Depends on:

- 1 Gtk+-2.6.4
- 2 libIdl
 - 2.1 Glib-2.6.3
- 3 zip

Mozilla-Firefox uses GTK+-1 by default and some special switches had to be passed to the configure script [35].

10.2.1. Configure switches

```
Export MOZILLA_OFFICIAL='1'  
export BUILD_OFFICIAL='1'  
export MOZ_PHOENIX='1'
```

```
./configure --prefix=/usr --enable-default-toolkit=gtk2
```

Mozilla-Firefox starts up much faster when compiled from source code than the precompiled packages.

10.3. ABI-Word

Depends on:

- 1 popt-1.7
- 2 libGlade
 - 2.1 libXml2
 - 2.2 Gtk+-2.6.4
- 3 fribidi

No problems were encountered while compiling ABI-Word.

10.4. BMP

Depends on:

- 1 libGlade
- 2 Gtk+-2.6.4
- 3 Glib-2.6.3
- 4 libvorbis (optional)
 - 4.1 libogg

No problems encountered while compiling Beep-Media-Player.

11. Appendix D - Data DVD

This DVD contains all of the software used in this project:

1. ISO files for Gentoo, Slackware and Mini-Slack
2. The LFS 6.0 Book
3. The complete source code for building LFS
4. Various data sheets for the SBCs used
5. Various articles
6. Savage4 Drivers for X.org 6.8.2
7. SC1200 OSS drivers
8. Geode ALSA, BSD and OSS drivers
9. Information on Gentoo and tweaking the compiler options
10. Beyond Linux From Scratch source code and installation instructions
11. The final Linux system (each directory tarred for convenience)
12. This documentation