SENIOR DESIGN PROJECT 2002

PERSONALIZED MULTIMEDIA JUKEBOX

April 20th, 2002

Advisor: Prof. Herr

Project Team:

Sam Fischbach
Mike Rykken
Andy Milby
Edie Bosse
# Table of Contents

Abstract ................................................................. p. 2

Introduction ............................................................. p. 3

Research and Gather Information ................................ pp. 3-4

- Existing Technology ............................................... p. 3
- Encoding Schemes ................................................. p. 4

Problem Definition .................................................... pp. 4-5

Develop a Plan .......................................................... pp. 5-6

- Fall Quarter ......................................................... p. 5
- Winter Quarter ...................................................... pp. 5-6
- Spring Quarter ..................................................... p. 6

Execute the Plan ....................................................... pp. 6-8

- Secondary Research ............................................. p. 6-8
- Listening Tests .................................................... p. 6-7
- Preliminary Hardware Requirements ........................ p. 7-8
- Preliminary Laptop Research ................................... p. 8

Real Jukebox Integration ............................................. p. 8-10

- Problems Encountered ........................................... p. 9-10
- Alternative Reevaluation ....................................... p. 10

Final Research .......................................................... p. 10-17

- Data Storage ...................................................... p. 10-12
- Wireless Data Access .......................................... p. 12
- Seamless Integration ........................................... p. 12
- Hardware Requirements ....................................... p. 12-15
- Laptop Research ................................................ p. 16

Final Configuration ................................................ p. 16-17

Final Implementation ............................................... p. 17-18

Verification ............................................................ p. 18-19

Documentation ........................................................ p. 19-20

Conclusion ............................................................. p. 20

Appendix A: Gantt Chart

Appendix B: System Tests

Appendix C: Media Jukebox User’s Guide

Appendix D: References and Reference Materials
Abstract

To get started, first a need was identified. This need was a Jukebox that is able to quickly input, store, and reproduce audio files from different media types (such as LP’s and CD’s). Next, after the need was identified, research and information gathering had to be done in order to understand what would be involved in providing the best answer for the need. This was carried out first by finding what products were already available that did some or all of what would satisfy this need. Further investigation found several standards that are commonly used in encoding audio files. To ensure that audio files would be encoded properly, several human, subjective listening tests were conducted to find the desirable quality of an audio file. Once enough data was gathered, a problem definition was written to clarify what the problem was and what direction would be taken in order to reach its solution. An essential part of any project is a plan structured with a time-line and that was done using the Gantt chart in Microsoft Project. A major discovery was stumbled upon while reviewing the different programs already available; a program was found that does all of the requested database functionality. Because of that, the time-line was significantly altered to reassign time to other areas of development. This jukebox program enabled the group to focus more on other aspects of the project without spending much time on the actual database. Time was spent downloading the music and information from records and compact discs, and also on developing what was soon to become a multimedia jukebox. Much of the effort involved product research to allow the group to purchase the necessary equipment at a reasonable price. Once the hardware was in the group’s possession, then it was a matter of implementing all of the equipment to come up with the final product. Final development involved preparing the jukebox, along with writing final documentation for the project.
Introduction

This year’s senior design project, advised by Professor Herr, was introduced with a presentation of a proposal for his group’s project. In this proposal he identified the need for which his group needed to design a solution. The need defined by Professor Herr was as follows: a jukebox was wanted that is able to quickly input, store, and reproduce audio files from different media types (such as LP’s and CD’s). It was also desired that this product be able to be implemented in a laptop computer or some other small-integrated device. Also, a server was to be implemented to allow a greater amount of information be stored. This server had to have a sufficient back-up storage plan so no information would be lost in the event of a computer crash. This was to be accomplished by RAID-5 database backup. The group decided to follow the six-step engineering design process to accomplish all of these results. This process was handed to our group and was explained as shown below.

0) Identify a Need
1) Research and Gather Information
2) Define Problem
3) Develop a Plan
4) Execute the Plan
5) Verification
6) Documentation

Research and Gather Information

Existing Technology:

In the initial information gathering stage, a list of current technology and software packages that might be used in the design was obtained. When looking for a specific software package we wanted to evaluate whether any would be compatible with the basic requirements of the project. A variety of database fields would be necessary for the varied amount of information that would need to be stored and searched. Another necessary component would be the ability to encode WAV files into compressed digital audio. Without this feature, another program would be needed to compress the audio. Otherwise, the large amount of storage needed for a jukebox using WAV files would not be practical. After checking the websites of various software packages, the decision was made to further test five packages. These packages can be seen in Table 1.
### Encoding Schemes:

The next area that had to be researched was the different formats available for compressing digital audio. A WAV file is the standard format for uncompressed digital audio. For practical purposes, this would be too large to store audio. To deal with this problem, a compressed audio format had to be found that has both retains high fidelity and also has a small enough file size to store a large music collection. Low bitrate encoding techniques are used in many applications to compress digital audio. Efficient compression is one of the basic goals of low bitrate encoding. This needs to be done with minimal loss to the original signal. The compression algorithm must also have fairly widespread acceptance so that any software packages chosen could implement this compression method and would be able to decode and play back the compressed data.

The first compressed format that was researched was MPEG-1. Work on this compression scheme was started in 1988 and completed in 1992.

MPEG-1 Layer-3 is more commonly known as mp3. This is the most complex out of the MPEG-1 layers. It is optimized for encoding at low bitrates with the most common bitrate for compression being 128 kbits/sec. The MPEG-1 format is considered to be a member of the hybrid filterbank class. A polyphase filterbank is used for all three layers of the MPEG-1. For mp3, a modified discrete cosine transform is used to aid in the compression.

Ogg Vorbis is a new encoding scheme that has just been introduced to the market within the past couple of years. Vorbis was created as an alternative to mp3 compression. It is an open-source compression format that offers the availability of easy upgrades. There are two parts to Ogg Vorbis. The first is Ogg, which is very similar to the AVI video format; it is a “container” for various media types. Vorbis is actually written inside the Ogg framework; it is the actual audio codec. Since Ogg Vorbis is an open-ended codec, it is free to download and use.

### Problem Definition

Once enough information had been researched the problem definition was defined as follows: to design a jukebox program that is able to properly encode audio signals from various media types, as well as take additional information and organize it in an extensive database. Audio must be acquired in raw digitized format from an analog or digital source. The raw digital audio would then be encoded into a compressed audio format of an acceptable quality. Once all the audio has been encoded and sorted it may then be

---

**Table 1: Software Packages**

<table>
<thead>
<tr>
<th>Package</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liquid Audio</td>
<td><a href="http://www.liquidaudio.com">www.liquidaudio.com</a></td>
</tr>
<tr>
<td>Media JukeBox</td>
<td><a href="http://www.medijukebox.com">www.medijukebox.com</a></td>
</tr>
<tr>
<td>Music Match</td>
<td><a href="http://www.musicmatch.com">www.musicmatch.com</a></td>
</tr>
<tr>
<td>Real JukeBox Plus</td>
<td><a href="http://www.realjukebox.com">www.realjukebox.com</a></td>
</tr>
<tr>
<td>Win Amp</td>
<td><a href="http://www.winamp.com">www.winamp.com</a></td>
</tr>
</tbody>
</table>
transferred to a yellow-book format audio CD. The database should include any relevant information about the music, such as categories for lyrics, artists, titles, cover art, side notes, track numbers, types of music, copyright dates, and for any other additional information. Next, hardware storage requirements for numerous audio files would have to be considered, once an estimate of the space required based on the compression scheme and database size were known. The final goal of this jukebox would be to implement the numerous hardware and software components into a laptop computer or some other small-integrated device.

**Develop a Plan**

After the problem definition was constructed, from studying the need defined earlier, a plan to reach the end product had to be developed. The engineering process is a cyclical process of research, implementation, verification, and documentation, narrowing and refining project concepts and goals with each iteration. Our first process was in developing the software for a database for our jukebox. The second process was in developing the desired encoding process. Finally, the last process was developing the hardware requirements for the jukebox system. After identifying these different engineering processes (which to some degree do overlap,) the plan for the project developed into a fall, winter, and spring quarter plan that can be seen in the Gantt Chart displayed as Appendix A.

**Fall Quarter:**

During fall quarter preliminary research, an inventory and a progress report were planned. The preliminary research that was done consisted of researching existing technology, recording parameters, recording methods, and desired listening preferences from listening tests. It was expected that the research on the recording parameters and recording methods would take the longest time, while the research of existing technologies was expected to be fairly quick in comparison. An inventory was required to know what had been collected as far as available technologies (i.e., jukebox software packages). Along with this list of jukeboxes packages, a decision matrix was created to make an objective decision on which jukebox software package would work, if any. Within the inventory, the list of jukeboxes took the greatest amount of time due to the complexity of compiling the numerous options and abilities of each program. The final step completed in our plan for fall quarter was a progress report. This report outlined and explained the entire process and eventual end product from this project.

**Winter Quarter:**

Following fall quarter, our plan for winter quarter included secondary research, development, implementation, and documentation. The secondary research for winter quarter planned to cover research on the hardware requirements and a continued development of the listening tests. The listening tests, which took about twenty minutes per individual, took much of the time for the secondary research; 40 individuals were asked to participate. Once all the research was completed, the development of the Multimedia Jukebox truly began.

To develop this jukebox, a hardware decision matrix was designed to make the best choices of hardware given our set of criteria. While it did not take as much time as expected,
much time was spent encoding the different media types so that they could be entered into the database. Next, the implementation of the Multimedia Jukebox was started. For that to be done, the hardware and software of the system needed to be linked together. That was expected to take up the most time during the implementation stage, due to the possibilities of bugs and other unforeseen problems caused by trying to link them together; an accurate assertion. Once that had been completed, the jukebox could be tested for the correct, desired operation.

**Spring Quarter:**

During the final quarter, spring, testing and final documentation were scheduled for execution. The final testing had been planned to finish the encoding of as much audio as we could get done and to construct the final design of the Multimedia Jukebox, i.e. the proof of concept. The construction of the final design was expected to take the longest to get done during spring quarter, due to the slow process of entering the data and records (LPs) into our database. To finish our project a final presentation is planned to take place at the end of spring quarter as can be seen in the Gantt Chart in Appendix A.

**Execute the Plan**

In order to execute this plan, ways were developed to take the music from the various sources such as CD’s and records and download them into the jukebox program. To accomplish this, hardware had to be found able to accommodate the Real Jukebox Plus program (see Table 2) and the storage of audio files. Although the jukebox was to be made to accept data from records, cassette tapes, compact discs, etc, this year’s project focused mainly on getting the compact discs into the program (although LPs were encoded with some success). In order to create a jukebox system on a single computer, we researched how much drive space will be needed to hold the project’s excess of 55,000 minutes of audio, and whether or not a laptop computer would be feasible for this project. In order to decide how much memory and drive space was needed, more research was performed concerning the listening preferences in recording the music, as better sound requires more drive space on the computer. Tests of sound quality for a variety of file formats were administered and the results factored into the final analysis. Popular encoding formats, such as WAV and mp3 were investigated. Newer media formats such as Ogg Vorbis, primarily created by the Xiphophorus Company and not yet as widely supported, are also being investigated. They each differ in implementation, file size, and subjective listening quality.

**Secondary Research:**

**Listening Tests**

Lastly, downloading an extensive amount of audio onto a computer is a challenging task. To have all of the music on the computer in standard WAV format is virtually impossible with today’s technology for economic reasons, and thus encoding for compression made its way into the project and greatly alleviated concerns over space requirements. There were a few different options when it came to encoding, such as MP3
files, Microsoft’s WMA, and the lesser-known format, Ogg Vorbis. Making the decision for the encoding scheme was a very important task that involved various tests. In order to make a clear decision that made for the best jukebox, we weighted the three categories in the decision making process and these categories are outlined below:

- Human Subjective Listening Tests (70%)
- Average File Size Considerations (20%)
- Encoding Schemes Support (10%)

As the listening test developed, we gained an understanding of what the listeners prefer as far as the different types of encoding went. We felt that this was by far the most important aspect because as some file types produced distorted or otherwise lower-fidelity sound. The average file size was not considered as important because of the developing technology and decreasing costs to buy a new computer with very large amounts of memory – any of the formats greatly reduced the space as required by WAVs.

**Preliminary Hardware Requirements**

In addition to the completion of the listening tests, research was done on the hardware requirements for the design of the jukebox. These included hard drives, a server case, a motherboard combination, RAM memory, and a laptop computer. The first of these areas addressed was the hard drives. A decision matrix was compiled from the data collected between the months of December through February, which can be seen in Table 3. The calculations in the hard drive decision matrix had the Maxtor Diamond Max 160GB hard drive as the best choice. Every hard drive considered used an EIDE interface. Although there were several different seek times for the hard drives, it was decided that their maximum difference of 1ms was not significant enough to affect any decisions made. Therefore, the rubric for this decision matrix was cost per gigabyte.

**Table 3: Hard Drive Decision Matrix**

<table>
<thead>
<tr>
<th>Brand &amp; Models</th>
<th>Capacity (GB)</th>
<th>Rotational Speed</th>
<th>Cost ($)</th>
<th>Cost Location</th>
<th>Cost/GB ($)</th>
<th>Number of Drives for 500GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGVM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC3L120AVVA07</td>
<td>120</td>
<td>7200 rpm</td>
<td>$321.32</td>
<td>Pricemonger 2</td>
<td>2.68</td>
<td>4.17</td>
</tr>
<tr>
<td>IC3L100AVVA07</td>
<td>100</td>
<td>7200 rpm</td>
<td>$255.73</td>
<td>Interpolation</td>
<td>2.56</td>
<td>5.00</td>
</tr>
<tr>
<td>IC3L080AVVA07</td>
<td>80</td>
<td>7200 rpm</td>
<td>$194.95</td>
<td>Pricemonger 2</td>
<td>2.44</td>
<td>6.25</td>
</tr>
<tr>
<td>Maxtor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diamond Max</td>
<td>160</td>
<td>5400 rpm</td>
<td>$292</td>
<td>Pricewatch</td>
<td>1.83</td>
<td>3.13</td>
</tr>
<tr>
<td>Ultra ATA/133</td>
<td>120</td>
<td>5400 rpm</td>
<td>$226</td>
<td>Pricewatch</td>
<td>1.88</td>
<td>4.17</td>
</tr>
<tr>
<td>Diamond Max Plus</td>
<td>100</td>
<td>5400 rpm</td>
<td>$217</td>
<td>Pricewatch</td>
<td>2.17</td>
<td>5.00</td>
</tr>
<tr>
<td>Seagate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST380021A</td>
<td>80</td>
<td>7200 rpm</td>
<td>$172</td>
<td>Pricewatch</td>
<td>2.15</td>
<td>6.25</td>
</tr>
<tr>
<td>ST380020A</td>
<td>80</td>
<td>5400 rpm</td>
<td>$179</td>
<td>Pricewatch</td>
<td>2.24</td>
<td>6.25</td>
</tr>
<tr>
<td>ST360020A</td>
<td>60</td>
<td>5400 rpm</td>
<td>$110</td>
<td>Pricewatch</td>
<td>1.83</td>
<td>8.33</td>
</tr>
<tr>
<td>Western Digital</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WD1200BB</td>
<td>120</td>
<td>7200 rpm</td>
<td>$275</td>
<td>Pricewatch</td>
<td>2.29</td>
<td>4.17</td>
</tr>
<tr>
<td>WD1000BB-SE</td>
<td>100</td>
<td>7200 rpm</td>
<td>$193</td>
<td>Pricewatch</td>
<td>1.93</td>
<td>5.00</td>
</tr>
<tr>
<td>WD800BB</td>
<td>80</td>
<td>7200 rpm</td>
<td>$163</td>
<td>Pricewatch</td>
<td>2.04</td>
<td>6.25</td>
</tr>
</tbody>
</table>
From research conducted during fall quarter, the team knew that about 500GB would be needed in total to accommodate Professor Herr’s current music collection while leaving room for calculated expansion. Basically, the calculation included the average file size produced by encoding to an Ogg file multiplied by the amount of material to encode (some 400+ CDs, etc) plus a “fuzzy factor” of around 30% to insure for future expansion needs.

We calculated that three of the selected hard drives, around 480GB, would be needed to fulfill this requirement. In addition, a fourth hard drive would be used as the back-up drive. This initial hard drive decision was completed on December 11. Research on the rest of the hardware was scheduled for later.

**Preliminary Laptop Research**

After the hard drive decision was made, research was done on laptops, including their current prices and possible configurations. Miscommunications within the team lead to incorrect laptop research. There were two basic categories to the research of the laptops. One category was the parts of the laptop that the design needed to be maximized for performance concern, and the other was the parts that needed to be minimized in order to reduce cost to the client. Examples of the parts that the design needed to be maximized are as follows: RAM memory and CD/RW drive. Examples of the parts that the design needed to be minimized are as follows: screen size, hard drive capacity, and processor speed. Mistakenly, the aspects that needed to be minimized were researched as if they needed maximized. All this information gathered was not of any use to the design and had to be thrown out. This setback put the team behind schedule, so we made the decision to move onto the process of integrating Ogg Vorbis into Real Jukebox and come back to the laptop research later.

**Real Jukebox Integration:**

When it came time to make a decision on a jukebox software program to use, a decision matrix was constructed to allow us to choose the best program suited for our design project. There were numerous topics involved in the decision process that we felt were meaningful with what we wanted to accomplish in our design, such as searchability, storage options, and encoding capability. Each of these topics was then assigned a value from a rubric, which described how well that software package operated at that individual task. The following rubric was defined to aid in the decision matrix:

- 0 - no option
- 1 – option is below average
- 2 – option is average
- 3 – option is above average

The decision matrix, using the rubric is shown below in Table 2.

**Table 2: Decision Matrix for Jukebox Software**
Initially, RealJukebox seemed to be the most promising retail package available. While RealJukebox did not natively support the Ogg Vorbis encoding format, RealNetworks, Inc., provided a System Developers’s Kit (SDK) to help alleviate such problems. This SDK included a code-base for writing 3rd party add-ons and plug-ins for all RealNetworks’ software as well as thorough documentation. Ogg Vorbis, which was released directly into the public domain, also has complete source code available, although it was not as well documented. As of the beginning of the project, development had already begun on creating an Ogg plug-in for RealJukebox, readily available at the Ogg website. According to the documentation, the plug-in was at the stage of playing back Ogg files, when compiled against the RealNetworks SDK. As was found out later, this was not entirely true.

Problems Encountered

After a preliminary delay of obtaining a suitable computer and the necessary development tools (namely Microsoft Visual Studio, etc) the entire source needed to compile the plug-in was downloaded. The provided Visual Studio project opened, but would not build. After a great deal of debugging, the problem was found to be a set of variables used one place in the Ogg source were never defined elsewhere. A new version of the Ogg source had recently been released, and the ‘side packages’ (that is, the non-core projects in the Vorbis development) had not yet been updated.

One of the interesting aspects of open source development and the massive growth of the internet in the last several years is the fact that one can easily contact the developers of a particular project via electronic means and often receive a very expedient response. The most famous example of this would be Linus Torvalds, the creator of Linux, who still controls the direction of development, talks over a Usenet newsgroup frequented by hundreds of programmers worldwide. In the case of this project, several of the developers of Ogg made themselves readily available on Internet Relay Chat (IRC) for real-time discussion while they were at work. In particular, one developer, “Jack”, was very helpful in diagnosing and resolving small problems we encountered, including the undefined variable problem above.

After the first issue had been resolved, several more arose. Linux is the primary development platform for Ogg, and while the core Ogg codec has been well distributed and tested across several platforms, less work has been done on the RealNetworks plug-ins, as RealNetworks software is available exclusively on MS Windows. As a result, there were some inconsistencies between the Linux and Windows project files resulting in compilation
errors. Eventually, with Jack’s help, all of the issues with the plug-in were worked out and a compilation was successful. However, it didn’t work.

The reason it didn’t work lay in the manner that RealNetworks had altered *their* code-base from the previous version used to develop the Ogg plug-in. The newest release of RealJukebox had required an extensive change in several of the programs sub-systems, which effectively broke compatibility with previous versions of the development tools, thus rendering the newly compiled plug-in useless. Jack contacted the people at RealNetworks in charge of development and determined that the plug-in code could be modified by adding multi-threading capabilities or re-written from scratch, either of which would take considerable development time. All work on the Ogg project is voluntary, the work of 4 or 5 main project leaders like Jack and several dozens of other people working in their spare time. Since the plug-in was merely a side project, Jack predicted that it would be unlikely that anyone would look over the several thousand lines of code it comprised anytime soon. As well, the RealNetworks SDK necessary to compile the Ogg encoder (the current plug-in only handled output) was not yet complete. Jack offered to continue helping in our efforts and also suggested that he could possibly finish the job for a fee, if we decided not to go ahead ourselves.

**Alternative Reevaluation**

At this time, it became necessary to reevaluate our options. Was the time or money that needed to be invested in this undertaking worth it, or would another option suffice?

We returned to our jukebox software matrix and decided to have another look at the next highest contender, Media Jukebox by the J River Corporation. Some time had passed since we had evaluated their software and a new beta version now supported most of the functionality we had felt previously that it lacked. Additionally, the developers were very open to suggestion and promised to help add new features we desired.

**Final Research:**

The server has three main purposes. The first and most important is to provide storage for the massive amount of expected data. The second is to provide wireless access to the data, so the laptop client doesn't have to be tethered by networking wire. The third purpose is to provide seamless integration to the laptop, making data access and storage transparent. The client provides the interface for all system functionality including encoding and playback of music.

**Data Storage**

Laptop technology has not yet advanced to the point of having the same storage capacities and higher storage densities needed. A separate server computer can contain a much greater capacity for data, thus eliminating the laptop storage deficiency. A top of the line laptop can be expected to have a 40Gb hard drive, at more than triple the price of a comparable desktop hard drive. In comparison, desktop hard drives come in capacities exceeding 160Gb - and 4 or more can be placed in a server case.
A second aspect of the server’s data storage is reliability. Tape backup, CD burning as backup, DVD burning as backup, and several other options were considered, but all were either not economically viable or would take too much time and too many resources. Software RAID-5 was chosen for its reliability, ease of setup, and economic practicality. RAID stands for Redundant Array of Independent Disks, and the 5 stands for the 5th of 5 possible modes of operation. In RAID-5, 3 or more disks are interlaced, with redundant data mirrored on each drive.

To the operating system, the array of disks appears as a single volume with free space of size* (N-1) sized disks.

For example, the 4 hard drives used in this project appear as 160Gb(4-1) = 480Gb. The last disk is reserved for a drive failure; when a disk in the array stops working, it is automatically removed from the array and the redundant information used to "rebuild" the lost disk on the spare. (See Figure 1)

![Figure 1: RAID-5 Setup and Implementation](image)

80Mb on each disk would also be allocated to a RAID-1 array. RAID-1 one mirrors the data across each drive, so there would effectively be 3 copies of the 80Mb partition. The partition would be used to store the boot-up information of the system, making it very difficult for the boot area to become faulty. Linux provides the functionality for all modes of RAID in the system kernel.

A third aspect of data integrity lies in the filesystem implemented. Unlike Windows, Linux supports a wide variety of filesystems, from the venerable MSDOS FAT16 filesystem to much more advanced, modern systems. One such filesystem, chosen for this project, is a data-efficient, journaling filesystem called ReiserFS. Unlike NTFS, commonly used on Windows servers, ReiserFS can pack several unrelated bytes into a hard drive block, increasing the density of data and gaining approximately 6% more usable disk space. Additionally, it is a journaling filesystem. Before the disk commits to writing data, a record
of when and where the data is being written is updated. In this way, if the power should fail, for example, the drive has a record of the last event that took place. Only a minimal amount of data would be lost and there would be no need to run a program like ScanDisk; the drive just picks up where it left off.

**Wireless Data Access**

The goal of this project was to provide a Jukebox system with a small, portable controller that is untethered by wires. The IEEE 802.11b wireless communication standard is the most prominent and widely used wireless implementation currently available and was therefore chosen for the project. A PCI wireless card was purchased for the server and a PCMCIA card for the laptop after research into the matter. As with most aspects of Linux, a small group handles development of a specific set of hardware drivers. In this case, the "Linux WLan" group (www.linux-wlan.org) is currently undertaking the development and improvement of 802.11b wireless networking under Linux.

The Linux WLan website has the latest drivers available for free download, which work for a variety of manufacturers' chipsets. Additionally, they also provide a mailing list for developers and users of their drivers.

The D-Link card chosen for the server, which runs the Intersil Prism2.5 chipset, was listed as being fully supported.

**Seamless Integration**

With data stored in one location and the Jukebox software running in another, a means of communication had to be devised between the two locations. The wireless device provides raw network access, but no more.

The system must be convenient and easy to use for even a novice computer user, so manual transmission of the data via ftp or another similar protocol would be impractical. Samba, another free open-source project, allows integration between UNIX and Windows platforms by 'fooling' windows into thinking a Linux server is actually a Windows machine. Samba allows one to setup a Windows share, much like and other Windows machine, complete with security features. In this case, the client machine then maps the foreign share to a local drive and to the Jukebox software, the share on the UNIX machine appears as nothing more than a local hard drive.

**Hardware Requirements**

The team decided to wait until two weeks before the design deadline, which is roughly the end of February or the beginning of March, in order to ensure accurate prices. First, the hard drive decision matrix was updated to include its most recent prices. This can be viewed in Table 4. The price from the preliminary research was $292 per hard drive, and the final research price had dropped down to $248 per hard drive.
Decisions then had to be made on the other hardware components of the jukebox server box. The second part that was researched was the motherboard combinations that were available. A new rubric had to be assigned to this data that would be the new rubric for the rest of the hardware researched for the server box. Each option evaluated was assigned a numeric value, from one point to three points. These points were then added up in each row and multiplied by the price of the component. This value was the score for each part that was considered, and the lowest score was the winner. The rubric was defined as follows: one point if the option was available and sufficient, two points if the option was available but wanted to be avoided, and three points if the option was non-existent. The motherboard combination decision matrix can be seen in Table 5. Processor speed was not a high priority, so cost could again be reduced. An option that was not commonly on the motherboard, which was needed by the design, was onboard video.

### Table 4: Final Hard Drive Decision Matrix

<table>
<thead>
<tr>
<th>Brand &amp; Models</th>
<th>Capacity (GB)</th>
<th>Rotational Speed</th>
<th>Cost ($)</th>
<th>Cost Location</th>
<th>Cost/GB ($)</th>
<th>Number of Drives for 500GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC35L120AVVA07</td>
<td>120</td>
<td>7200 rpm</td>
<td>$260.00</td>
<td>Pricewatch</td>
<td>2.17</td>
<td>5</td>
</tr>
<tr>
<td>IC35L100AVVA07</td>
<td>100</td>
<td>7200 rpm</td>
<td>$202.08</td>
<td>Interpolation</td>
<td>2.02</td>
<td>5</td>
</tr>
<tr>
<td>IC35L080AVVA07</td>
<td>80</td>
<td>7200 rpm</td>
<td>$150.00</td>
<td>Pricewatch</td>
<td>1.88</td>
<td>7</td>
</tr>
<tr>
<td>Maxtor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diamond Max</td>
<td>160</td>
<td>5400 rpm</td>
<td>$248</td>
<td>Pricewatch</td>
<td>1.55</td>
<td>4</td>
</tr>
<tr>
<td>Ultra ATA/133</td>
<td>120</td>
<td>5400 rpm</td>
<td>$192</td>
<td>Pricewatch</td>
<td>1.60</td>
<td>5</td>
</tr>
<tr>
<td>Diamond Max Plus</td>
<td>100</td>
<td>5400 rpm</td>
<td>$195</td>
<td>Pricewatch</td>
<td>1.95</td>
<td>5</td>
</tr>
<tr>
<td>Seagate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST380021A</td>
<td>80</td>
<td>7200 rpm</td>
<td>$138</td>
<td>Pricewatch</td>
<td>1.73</td>
<td>7</td>
</tr>
<tr>
<td>ST380020A</td>
<td>80</td>
<td>5400 rpm</td>
<td>$125</td>
<td>Pricewatch</td>
<td>1.56</td>
<td>7</td>
</tr>
<tr>
<td>ST360020A</td>
<td>60</td>
<td>5400 rpm</td>
<td>$94</td>
<td>Pricewatch</td>
<td>1.57</td>
<td>9</td>
</tr>
<tr>
<td>Western Digital</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WD1200BB</td>
<td>120</td>
<td>7200 rpm</td>
<td>$197</td>
<td>Pricewatch</td>
<td>1.64</td>
<td>5</td>
</tr>
<tr>
<td>WD1000BB-SE</td>
<td>100</td>
<td>7200 rpm</td>
<td>$190</td>
<td>Pricewatch</td>
<td>1.90</td>
<td>5</td>
</tr>
<tr>
<td>WD800BB</td>
<td>80</td>
<td>7200 rpm</td>
<td>$142</td>
<td>Pricewatch</td>
<td>1.78</td>
<td>7</td>
</tr>
</tbody>
</table>

### Table 5: Motherboard Combination Decision Matrix

<table>
<thead>
<tr>
<th>Brand</th>
<th>Case</th>
<th>Power Supply</th>
<th>Fan/Heat Sink</th>
<th>Motherboard</th>
<th>Processor</th>
<th>Onboard Video</th>
<th>Price</th>
<th>Total Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>AMD K6/2 533 -1</td>
<td>1</td>
<td>82</td>
<td>984</td>
</tr>
<tr>
<td>Biostar</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>Celeron 533 -2</td>
<td>3</td>
<td>79</td>
<td>1185</td>
</tr>
<tr>
<td>Generic</td>
<td>3</td>
<td>3</td>
<td>fan only -2</td>
<td>1</td>
<td>AMD K6/2 475 -1</td>
<td>1</td>
<td>79</td>
<td>869</td>
</tr>
<tr>
<td>Biostar</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>83</td>
<td>996</td>
</tr>
<tr>
<td>ECS</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>AMD duron 800 -1</td>
<td>3</td>
<td>82</td>
<td>984</td>
</tr>
<tr>
<td>Biostar</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>AMD duron 850 -1</td>
<td>1</td>
<td>86</td>
<td>860</td>
</tr>
<tr>
<td>Biostar</td>
<td>3</td>
<td>3</td>
<td>fan only -2</td>
<td>1</td>
<td>AMD duron 850 -1</td>
<td>1</td>
<td>87</td>
<td>957</td>
</tr>
</tbody>
</table>

The next hardware component researched was the RAM memory that the design would need to handle the database and the music files. The team figured that the system should have at least 256MB of memory; to handle the overhead of software RAID, ensure high performance, etc. This gave the team the option to go for two 128MB chips or one 256MB chip. The decision of the memory can be seen in Table 6. The same rubric was used for the memory that was used for the motherboard combinations. Careful consideration of the pin organization had to be taken according to which motherboard combination was chosen.
The fourth hardware component to be researched was a server tower to be the case for the server box of the jukebox. A power supply was needed with the case in order to supply power to the system components. In addition, fans were considered in this decision. The decision matrix for the server tower can be seen in Table 7. Most server cases came with a 250W power supply, but since our design required four hard drives a bigger power supply was needed to make sure sufficient power would be available. As seen in Table 7 a 300W power supply or greater was researched.

<table>
<thead>
<tr>
<th>Brand</th>
<th>Bus</th>
<th>Size</th>
<th>Price</th>
<th>Total Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>PC 133-1</td>
<td>128-2</td>
<td>for two - 38</td>
<td>114</td>
</tr>
<tr>
<td>Generic</td>
<td>PC 100-2</td>
<td>128-2</td>
<td>for two - 38</td>
<td>152</td>
</tr>
<tr>
<td>House Brand</td>
<td>PC 133-1</td>
<td>128-2</td>
<td>for two - 40</td>
<td>120</td>
</tr>
<tr>
<td>Generic</td>
<td>PC 100-2</td>
<td>256-1</td>
<td></td>
<td>105</td>
</tr>
<tr>
<td>House Brand</td>
<td>PC 133-1</td>
<td>64x4-256-1</td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>Generic</td>
<td>PC 133-1</td>
<td>32x4-256-1</td>
<td></td>
<td>72</td>
</tr>
</tbody>
</table>

The fifth and final hardware component researched was the wireless networking devices that would be needed to communicate between the laptop and server. This was a new idea proposed to the client late in the design. The client approved the idea and a rush on the research had to be done. Originally, the laptop was going to have to be wired to the server to accomplish data transfers. With the addition of the wireless network, the design of the system became much more versatile. Now the user could carry the laptop with him wherever the user wanted to listen to the jukebox, and the server could remain stationary with the external speakers. The decision matrices for the wireless components are labeled as Table 8: subdivided into Table 8A, Table 8B, and Table 8C for clarity. The research for the wireless networking was completed on February 22. From these tables the D-Link for the laptop and the server are the best choices.

<table>
<thead>
<tr>
<th>Brand</th>
<th>Power Supply</th>
<th>Fan</th>
<th># of Bays</th>
<th>Price</th>
<th>Total Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>400W-1</td>
<td>3</td>
<td>7. - 1</td>
<td>38</td>
<td>190</td>
</tr>
<tr>
<td>Generic</td>
<td>300W-1</td>
<td>3</td>
<td>7. - 1</td>
<td>49</td>
<td>245</td>
</tr>
<tr>
<td>Generic</td>
<td>300W-1</td>
<td>2 80mm fans - 1</td>
<td>7. - 1</td>
<td>49</td>
<td>147</td>
</tr>
<tr>
<td>Codegen</td>
<td>350W-1</td>
<td>2 8cm fans - 1</td>
<td>only 5.25&quot; - 6. - 2</td>
<td>50</td>
<td>200</td>
</tr>
<tr>
<td>Foxconn</td>
<td>350W-1</td>
<td>3</td>
<td>7. - 1</td>
<td>61</td>
<td>305</td>
</tr>
</tbody>
</table>

Table 8A: PCMCIA Wireless Decision Matrix
The wireless networking data was researched on the possibility of using an Access Point (AP) for communication between the laptop and the server. The cost of an AP varied greatly, from $120 up to over $1000, depending upon the number of network devices it could communicate with. Upon learning about the purpose of the AP, the team decided that cost could be reduced by not using an AP. This decision was discussed with the client and he verified that the network was only going to communicate between the jukebox laptop and the jukebox server. Therefore, the AP was not needed for the design unless other devices were added later. In that case the client has the team’s research on the AP in Table 8C.

As the decision matrices on the wireless components indicate, the D-Link components were the best choice. The D-Link components operate in 2.4 GHz Direct Sequence Spread Spectrum (DSSS). These components use a 128-bit wired equivalent privacy (WEP) encryption method in order to secure information sent from the laptop to the server. There are many wireless components that are still using 64-bit encryption. All the components researched were compliant with the IEEE 802.11 standard. This standard sets the transmission rates that depend on the distance from one communicating device to another communicating device. Transmission rates operate optimally at 11mbps, then fall to 5.5mbps, then to 2mbps, and finally down to 1mbps. This wireless network has been rated for communication indoors up to 328 feet and outdoors up to 984 feet.

**Laptop Research**
After the research on the server parts was completed, except for the wireless networking devices, the team renewed the research on available laptops that met the requirements mentioned earlier in the Preliminary Laptop Research section. Many laptop companies were considered, but only those laptops that had a CD/RW drive were considered. The team was looking for the maximum amount of RAM, but 128MB and up would be sufficient. All the laptops researched had hard drives with enough capacity to run the database program, so the hard drives didn’t affect the decision. The decision matrix for the laptops can be seen in Table 9. The best choice on a laptop, from Table 9, was the Compaq Presario 700. This was an economically good choice for the client because it was inexpensive compared to the other available laptops, and it met all the requirements needed in order to comply with the design of the jukebox system.

Table 9: Laptop Decision Matrix

<table>
<thead>
<tr>
<th>Brand</th>
<th>Model</th>
<th>Processor</th>
<th>RAM</th>
<th>Hard Drive</th>
<th>Screen</th>
<th>CD-RW</th>
<th>Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP</td>
<td>Omnibook xe31</td>
<td>Celeron 866MHz</td>
<td>128MB - 2</td>
<td>10GB - 1</td>
<td>14.1&quot; XGA TFT - 1</td>
<td>8x4x24 - 1</td>
<td>1039</td>
<td>7273</td>
</tr>
<tr>
<td>Toshiba</td>
<td>Satellite 1000-S157</td>
<td>PII 1.06GHz</td>
<td>256MB - 1</td>
<td>15GB - 1</td>
<td>14.1&quot; TFT - 1</td>
<td>8x4x24 - 1</td>
<td>1223</td>
<td>6115</td>
</tr>
<tr>
<td>Compaq</td>
<td>Presario 700</td>
<td>AMD 1.0GHz</td>
<td>128MB - 2</td>
<td>10GB - 1</td>
<td>13.3&quot; XGA TFT - 1</td>
<td>8x4x24 - 1</td>
<td>1009</td>
<td>6054</td>
</tr>
<tr>
<td>Gateway</td>
<td>Solo 5350LS</td>
<td>PII 1.06GHz</td>
<td>256MB - 1</td>
<td>20GB - 1</td>
<td>14.1&quot; XGA TFT - 1</td>
<td>8x4x24 - 1</td>
<td>1499</td>
<td>7495</td>
</tr>
<tr>
<td>Dell</td>
<td>Inspiron 2500</td>
<td>Celeron 900MHz</td>
<td>128MB - 2</td>
<td>20GB - 1</td>
<td>12.1&quot; SVGA TFT - 1</td>
<td>16x - 1</td>
<td>1167</td>
<td>8169</td>
</tr>
</tbody>
</table>

Final Configuration:

The Jukebox system consists of two main parts: a client and a server. The server hardware was assembled from the following, as discussed above:

- **CPU:** AMD Duron, 850Mhz
- **Motherboard:** Biostar M7-KVS (Integrated Video)
- **RAM:** 256Mb PC133 168 Pin Dimm
- **Hard Disk(s):** 4 x 160Gb, Maxtor 5400rpm Drives
- **PCI Cards:** D-Link DW-520 PCI 802.11b Wireless Card
  - Promise Ultra100 ATA HDD Controller
  - Netlink 10/100 Ethernet card (for debugging)

The client configuration:

- **CPU:** AMD Mobile Duron, 1Ghz*
- **RAM:** 128Mb
- **Hard Disk(s):** 10Gb
- **CDRom:** 8x CD Writer
- **Sound:** Integrated Sound plus 1/8" Stereo Jack
- **PCMCIA:** D-Link DW-650 PCMCIA 802.11b Wireless Card

Mandrake Linux 8.2 was chosen for the server operating system, due to it being a highly up-to-date, free distribution of Linux. The OS kernel version was 2.4.18, released in March of 2002. The laptop shipped with Windows XP.
Final Implementation:

The laptop arrived first. Media Jukebox was installed and tested begun on all of the features that could be tested before the system was complete. The rest of the parts arrived and the server was then assembled. Mandrake Linux was installed on the server and work begun on the system.

The first task that needed to be accomplished was to determine that the wireless networking would work properly. The latest Linux-WLan drivers, 0.1.13 (February), were downloaded and compiled without error. They installed correctly and after several hours of attempting to configure the card, a small file was sent from the server to the laptop via ftp.

Seeing that wireless networking was working, then next step was to move on to the RAID-5 setup. This would be more difficult, as the method that would be used is complex and not forgiving of errors. Since all four disks would be used in the RAID array, and one of them already contained the operating system, it would be necessary to create the raid array and then move the live file system to it in one pass before rebooting. This is akin to moving Windows from one disk to another while it is running and then pulling the first disk from the system.

An online tutorial provided a step-by-step guide on how to accomplish this feat. Although it was followed, the process never worked without problems. After approximately 60 hours of debugging, the system was wiped clean and begun anew, with a great deal more knowledge of the inner working of RAIDys. Within several hours, the RAID-5 array was up and running on the clean system. The redundant RAID-1 boot-area was not implemented due to problems previously encountered with the system not starting. Instead, 2 Gb on each disk was set aside for the operating system and each disk made bootable. In the event of one of the disks failing, one would merely have to change the disk boot-order in the bios, as opposed to RAID-1's automatic switching from the defective disk.

The next priority came in restoring wireless networking to the server. Although a new version of the Linux-WLan drivers - 0.1.14 - had come out, 0.1.13 was kept, as it was known to work; the old adage says not to fix what's not broken. The drivers again compiled and installed without problem.

Everything was proceeding well. Only the last step remained, configuring Samba to allow transparent file sharing. For test purposes, a completely open and insecure Samba share was created. On the laptop, the "JukeServer" appeared in the Windows networking box and it had one share, which the laptop had no trouble accessing. However, after several minutes, "JukeServer" disappeared. Both the server and the laptop insisted that wireless networking was operational. After several days of debugging various configuration files and checking all of the system log files, the cause of the disappearing network connection was traced to the wireless card installed in the server. Further inquiry revealed that other users of the same card had been reporting similar problem, from late February through April. No solution had been found and most people in the Linux-WLan mailing list were perplexed by the problem. The card itself, the DWL-520, was identical in most respects.
to the reference design offered by chipset manufacturer Intersil, the design used in several other working PCI cards. The newer, 0.1.14 drivers proved to be of no help.

Some recommendations were given about how the card might be made to work, but none panned out. It appears that our only major obstacle to a complete working system lies in installing a different wireless adapter on the server. All other components appear to work.

**Verification**

The verification process started with the initial installation of the Media Jukebox program. After the installation was completed a series of tests have been run to see if this program met our specifications. These tests were included in Appendix B.

The initial testing of the system included a series of tests involving the encoding and format of the various media files. A list of included media types with the Media Jukebox program will be included in Appendix D; the designers of Media Jukebox give this list.

The first test was to record from a CD. Encoding a list of songs from a CD to both 128 kbps and 256 kbps first tested the MP3 codec. These tests passed by properly encoding and maintaining the necessary fidelity of the music. The next tests were of with the Ogg-Vorbis format, which was used for the design. Ogg was tested at 128 kbits, 256 kbits, and 350 kbits. Each of these tests done with Ogg Vorbis also passed our requirements. A test involving the encoding of music to an uncompressed WAV file was completed.

The next series of tests were run to determine if one format could be converted to another if was needed. The base WAV file of one of the songs was taken and converted to Ogg Vorbis and MP3 at 128 kbits and 256 kbits. All of the tests involving this conversion from a WAV file to another format were flawless. The conversion from one compressed format to another, such as MP3 to Ogg, was not tested. These tests were not done due to the undesirable degradation of sound that would occur from the conversion of one compressed format to another.

A series of tests were also conducted involving the use of an external record player to encode from records. The records were recorded into Ogg Vorbis format at 256 kbps. A series of records were used to see if this process could be completed on a series of different recordings. Initial attempts to record showed that some adjustments had to be made to record from the device. In order for the individual tracks to be identified by Media Jukebox, a setting in the program was used to determine the empty space between two tracks. The default settings were determined to be of greater length than the silence between the tracks of the records. The amount of time between the tracks was determined through trying a shorter time, and this process was repeated until the recording process successfully identified the end of one track and the beginning of the next track. The best time was found to be a setting of 1200 ms.

Some other issues that had to be tested involved the database entries and database search capabilities. The information that was needed to be stored for each album had already been determined. The software was then checked against the necessary fields to determine if all of the fields were included. The software included, as part of the database field, a link to
a website for further information. This effectively makes the possibilities of expanding the
database limitless. Since the Linux server has web page hosting software, custom html can
be written and placed on it. The next issue was the database search capability. In order to
test this, a small amount of music was loaded into the media library. The software has a
search bar at the top of the media library to allow for quick and easy searches. The tests were
completed and proved that the search capability of the database is more than adequate.

The last issue that was addressed was the hardware configuration. The laptop portion
of the design has been implemented and fully verified for functionality. The server side of
the design is currently still being tested and verified. Mandrake Linux is being used as the
operating system of the server. This OS allows us to use a Raid 5 system for backup for the
480 GB of storage. This system involves an immediate backup if one of the drives fails. We
have used three 160 GB hard drives for storage and one 160 GB hard drive for backup. This
system of backup has been tested and verified with the “simulated failure tools” provided.
The system that is currently being implemented and tested is the wireless network
connection. When this is verified the laptop and server will be connected via a wireless
network. The laptop will be able to use the 480 GB of hard drive space for storage of the
music files. The hard drives will be shared as a Windows directory through the Linux
implementation of ‘Samba’. Samba implements the stored drives in a Linux server as a
single shared Windows drive over the network. Although the system has worked completely
for a short amount of time, issues with the wireless PCI card on the server prevented us from
declaring the system finished. Once the card has been replaced, the system should work as
desired.

**Documentation**

For each phase of the project, a written record has been kept which details the
activities of each group member with regard to their individual subprojects. This included
minutes of group meetings, information provided by each member, and documentation
gathered by the group as a whole.

For each system component, specifications, technical information, manuals, and other
data have been compiled and archived for later reference. A progress report was produced
containing information on the status of the project, decisions made, and references to data
gathered.

After the project was completed and verified, a complete archive of all information
acquired during the course of the project was assembled and a comprehensive compilation of
pertinent end-user information was gathered in the form of a systems manual.

**Conclusion**

As soon as the guidelines and expectations for this years senior design groups were
announced, there was immediate action taken to ensure that this project was to be a success.
Fall quarter was filled with preliminary research on all of the project requirements that were
given by Professor Herr. A Gantt Chart was produced to guide the design team. It laid out
what was to be done and when it was to be done. This enabled the group to maintain a
working environment, and to not lose focus of the overall goals. The listening tests that were
developed were a necessary task to decide on the proper compression format. The project came full circle during the winter quarter, as the jukebox software was selected, and music and information downloading had begun. The research part of the project was not completed during winter quarter, as investigation into the proper hardware was an integral part of the project. Once items such as the laptop, server, and backup database were selected and purchased, the remaining task for the spring quarter was to make sure that all of the individual pieces to the puzzle would fall into place. This was accomplished by the group’s knowledge of computers and their components. When the project was initially started, there was a list of ideas and needs given to the group members, and with persistence and hard work, all of the group’s goals were accomplished.