Media Center oriented Linux Operating System

Tudor MIU, Olivia STANESCU, Ana CONSTANTIN, Sorin LACRIȚEANU, Roxana GRIGORE, Domnina BURCA, Tudor CONSTANTINESCU, Alexandru RADOVICI

University "Politehnica" of Bucharest
Faculty of Engineering in Foreign Languages

tudor.miu@gmail.com, olivia_uk@hotmail.com, constantin.ana89@gmail.com, sorin_lacriteanu@yahoo.fr, roxana_grigore89@yahoo.com, domnina_burca@yahoo.com, tudormihai_constantinescu@yahoo.com, , msg4alex@gmail.com

Abstract: Nowadays there is a high demand of computer controller multimedia home systems. A great variety of computer software media center systems is available on the market, software which transforms an ordinary computer into a home media system. This means it adds some functionality to the normal computer. Our goal is to develop such a user-friendly intuitive system, dedicated for home media centers. In contrast with other proprietary approaches (Windows Media Center, Apple TV), we are building an entire operating system specialized on this. It is based on the Linux kernel, thus providing high portability and flexibility at a very low cost. The system is designed to work out of the box (*plug it in and use it*), needing zero configurations (no human configuration as much as possible) and no installation (Linux-live system, works from a CD, DVD or USB device).

The user interface is not more complicated than a generic TV user interface. In this aim, the file system is hidden from the user, being replaced with an intuitive media library, the driver configurations is done automatically, network configuration is also handled without user actions (as much as possible).

Key-words: OS, media center, Linux, multimedia, portable, intuitive, free, open-source

1. Introduction

The first Linux system was created within an academic environment. With a goal of obtaining a free alternative for commercial UNIX systems at that time, members of an enthusiast community gathered around Linus Torvalds in order to help improving and extending *the kernel*, kernel that he initially created just as a university project. Subsequently, plenty of software projects based on this kernel have been founded, thus giving birth to Linux distributions. Their importance grew more and more each year, achieving nowadays 2% of the market.

Although Linux distributions are free (and some aren't subject to copyright law), they require advanced operating systems knowledge and skills due to the simple reason of lack of graphical interface tools. The existing ones are quite difficult to be handled by those who don't have a programming background. Just imagine an unskilled user trying to set up the sound modules for the *kernel*. The user doesn't know exactly what the sound card is; nevertheless would he know what a *kernel module* means.

Even advanced user interfaces like KDE and GNOME (which by the way occupy a lot of disk space) cannot be considered intuitive. If for us, it takes several minutes just to find the configuration tools, imagine a user which has just bought a computer and wants to watch some movies. This is exactly like the comparison between a TV and a computer. The user goes to the store, buys a TV, plugs it in and it works. On the other hand, if he buys a computer, he will plug it in, will have to wait … and wait … and most surely reboot it (*what does that mean ?*) and maybe than it will display something. In order to actually watch some movies or listen to some music, downloading application is necessary. The result: the user sticks to the TV. [TW06, Ta04]. This means that serious improvement in computer user interfaces is needed.

When it comes to software for playing multimedia, the sky is the limit. Players like Winamp, Windows Media Player, MPlayer [MT08], Amarok, XMMS, GeeXboX, Domotixi, MMS are just a small list of examples. The problem: they lack an attractive user interface, they are not intuitive to use and usually don't meet all of today's users' most common needs. For sure, an ordinary hardware DVD player will be the best choice for a *computer unskilled* user.

Further on, we will present some related work, meaning other Linux approaches to multimedia [CM08].

GeeXboX is a free embedded Linux distribution which aims at turning the computer into a so called HTPC (Home Theater PC) or Media Center. Being a standalone LiveCD-based distribution, it's a ready to boot operating system than works on any Pentium-class x86 computer or PowerPC Macintosh, implying no software requirement. It supports playback of nearly any kind of audio/video and image files and all known codecs and containers are shipped in, allowing playing them from various physical supports, either being CD, DVD, HDD, LAN or Internet.

Domotixi is a free Media Center application for Windows that gathers under the same interface multimedia, communication functions and access to information. It enables the user to watch TV, listen to CDs, Mp3s, watch movies and access information about it (actors, reviews etc), listen to radio, consult the weather forecast, and communicate with MSN contacts. Its interface is adjusted so it can be used on a TV and controlled with the help of a remote controller.

MMS (My Media System) is an application that manages, displays and plays media content such as videos, music, pictures, and more. MMS runs perfectly on anything from a Set-Top-Box connected to your TV-Set, to your specially tailored multimedia PC and HD display. As the name implies, it is a media

system *with the user in control*. It lets other applications such as MPlayer, VDR, or Xine take care of what they respectively do best, and integrates them into one system, that is easy to understand and operate.

The goal of this project was to create a simple, intuitive and easy to use Linux based media center. This has been achieved by designing a system which starts from a CD, DVD or USB stick (thus does not require any installation), tries to automatically configure the hardware devices (video, network, etc.), hides the file system from the user by means of a media library and provides an intuitive graphical user interface.

In order to test our goal achievement, we have provided the system to people that do not have any knowledge of computers, so that we could observe their reaction. We will also provide some public beta testing in the future in order to improve usability. Moreover, we will try to detect the most used configuration patterns, so that we can apply them to the new versions of the system.

2. Description

Multimedia oriented system

The aim of this project was to create media center Linux distribution. Unlike other commercial products that exist on the market, which are basically only some software upon a normal general purpose operating system, this project's sole purpose is being a media center. This means that no other software tools are installed (browsers, test editors, etc.). The user interface was specifically design for multimedia (no *Start Menu* or *Applications Menu*, no icons files managers, etc.).

Moreover the system's size has been dramatically reduced, so that it can fit on a very small space. All the other available space on de the device (CD, DVD or USB stick) will be used for multimedia material (music, videos, pictures, etc.).

Several notable differences from a general purpose system are worth mentioning. First of all, the system tries it's best to auto configure the device drivers. This means it will try applying several *known good configurations*. The file system is hidden from the user. Usually file systems can be tricky and have no sense on a multimedia system. Organizing multimedia content into a media library is a much better approach. While searching songs or videos, users prefer searching through a library than through folders and files. For this, several *intelligent* multimedia scanners are used.

The user interface is unique. No Linux standard user interface (such as KDE, Gnome, Xfce, etc.) is used. The main idea of the design is *three-clicks-away - if some action takes more than three clicks from the user, it must be redesigned*.

Minimal Live Operating System

In order to use Linux, a minimal live[1] system had to be setup. We have used the *Linux From Scratch* [Be08] (further referred as LFS) approach. This means compiling a minimal Linux system, which should be able to run any program without a problem. This is good, but not enough. The system compiled like this does not make any configuration automatically. Several tools for auto configuring had to be developed. Let's discuss them a little.

Distribution

The aim of the project is to create a mobile drive (CD, DVD or USB stick) with a very small Linux system and a lot of space for multimedia material. Once plugged in, this mobile drive should be able to boot and transform the computer into a dedicated media center. It should be able to combine the multimedia material located on the drive with the multimedia material that if finds on the host computer. Thus, one actually obtains a *portable media center*.

The software for this distribution is minimal: the *kernel*, *startup and auto configuration tools*, *graphical user interface* (the media center), *media library* and a *media assistant*.

Kernel

One of the most important parts is the Linux kernel. After compilation, any operating system kernel does not do very much. It needs device drivers (*modules*) in order to be able to communicate with hardware devices. This is somehow tricky in the case of Linux systems, as default kernel modules are mostly generic. Companies usually don't write hardware code for Linux (e.g. Wi Fi card drivers) and/or release it under licenses different from GPL (e.g. NVIDIA Graphics Card driver). Our approach is to compile a non GPL kernel, a kernel which contains proprietary drivers.

The drivers that we have tried to include are *NVIDIA graphics card drivers*, *ATI Graphic card drivers*, *Intel graphics card drivers* and *Intel Wi Fi network drivers*. These are still undergoing tests, with some success for now. Even without these drivers, the system is still able to run graphics applications, using generic drivers. Performance is lower though.

Another important part of a media center is the sound subsystem. This usually works with the *kernel* generic drivers, but with lower performance. Sadly, there is not much that we can do, as most of the soundcards found on the systems are *on-board*, which means *software cards*[2]. The main problem with software cards is that the 5.1 and/or 7.1 sound systems are specific for every card, so it is not working with generic drivers.

Network is another key point for our media center. Even if the system is not a

[1] A live Linux system is a distribution which boots from a mobile device (CD, DVD, USB stick etc.) and runs without installing anything on the host computer. Examples of live distributions are: *Knoppix*, *ROSLiMS*, *SuSE Live*, *Mandriva One*, *Ubuntu installation*.

[2] Software sound cards are completely software controlled, and without hardware specification, no drivers can be written.

general purpose one, network and Internet connections are important. People might want to share files or download multimedia content. As simple as it might sound, configuration of network devices is a real problem. For now, we are focusing on wired connections (connection which the kernel automatically recognizes). Wi Fi connections are supported only for *Intel Centrino* systems, which are more or less easily recognized by the kernel.

Startup and auto configuration tools

These are key software tools for our media center distribution. The major problem of Linux systems is configuration. As powerful as command line configuration tools are [GA06, So05], as hard to use they get. Most of the times it is rather difficult for experienced users to configure Linux systems. Imagine how it is for inexperienced ones. We are getting back at our *TV problem*[3] [TW06, Ta04].

We have designed special startup software which will do it's best to configure the system. First of all we start with the hardware devices. We try to detect which hardware devices are on the system and subsequently load the respective kernel modules. If no suitable module is found, the generic modules will be loaded as a fallback solution. The errors will be logged so that they can be presented to the user[4].

Once the device drivers are loaded, the next step is to start the graphical subsystem. This poses some problems, as the configuration of the graphical environment has to be done dynamically. Remember that our goal is to start the system on every configuration, meaning different kind of hardware. Our auto configuration tool will do it's best to test the hardware and configure the environment. Upon success, the startup tool will launch the media center graphical user interface.

As a fallback solution in case of graphical system failure, the media center provides a text mode user interface. This is not as shiny as the graphical one, but still provides access to the multimedia content[5].

With the modules and user interface loaded, the auto configuration tool can begin configuring other parameters which can be done while the user already uses the system. We are referring here to *network configuration*, *multimedia content scanning* and other non key configuration points.

Network configuration is somehow tricky. The simplest way is if the user has access to a DHCP network. In this case, the system will be provided automatically with the configuration needed. The only problem is in setting necessary routing parameters, if more than one connection exists (e.g. wired connection together with a Wi Fi one). This is done fairly easy, with no difficulties. Our tool also tries to test the network connection for connectivity.

---

[3] The comparison between a user buying a TV and a computer, the TV works out of the box, the computer doesn't.

[4] The user will be informed about problems by means of a *media assistant*.

[5] Video multimedia content might not be available, depending on the graphics driver problem.

If no DHCP service exists, the configuration tool will try to set up some *well-known* configuration. This means setting up *classical* IP, DNS and routing schemes (e.g. the most used IP address for the gateway is 192.168.0.1 and for the stations 192.168.0.x The DNS is usually the same as the gateway). This is a brute force trial and error method, which in most cases works perfectly. If none of these works, the error will be logged and the user will be asked for input.

The multimedia content of the host computer will be also scanned. It is somehow awkward to ask the user to find all of his media files on the system. A special tool will scan all the hard drives of the computer and progressively add the found content to the library. This is done completely transparent for the user, user which will only notice that more and more media files appear in the library.

Special heuristics is used to classify media content. This is needed as not all the files on a computer a *real media files*. Imagine for instance that if *Microsoft Windows* is installed on the host computer, several *wav* files will be found on the hard drives. These files contain sounds used by the *Windows* graphical interface. Obviously, they must be filtered out. Moreover, images that are used for creating the interfaces of several programs are also on the hard drive (e.g. messengers' emoticons).

Various algorithms are still under testing, so no conclusions can be drawn yet regarding the efficiency of the algorithm.

Last but no least, an important function is the *plug and play device detection*. This tool detects automatically the insertion of portable devices and adds their contents to the media library. This is a main function for the user which simply wants to insert a camera card and watch his photos or a music player and listen to the music from it[6].

Graphical User Interface

Standard Linux graphical interfaces (GUI) are not suitable for the media center. They are general purpose GUIs, which have much more functions than needed. Moreover, they occupy a lot of space and, sadly, are not very intuitive.

We have designed a completely new user interface, as stated before, guided by the *three-clicks-away* principle. The GUI should be as simple as possible, but as useful as possible. The user must be able to reach any function as fast and as intuitive as possible.

The GUI focuses on three main functions: playback of music, videos and display of photos. We will discuss about all of them.

Photos

When using a media center, a user wants to be able to view his photos. This

[6] Due to propietary formats and digital rights protections, compatibility with Apple iPod is not provided.

function of the GUI allows the user to browse through his photos, view slideshows, and organize them. Future work hopefully will also provide image modification tools.

Music

The main component of a media center is the music system. Mainly media centers are used to playback music. Functions like play lists, next songs[7], repeat, and shuffle are standard for any player. In additions to these, we provide intelligent play lists, based on user ratings, play rates and other parameters. We are still testing several algorithms which will be described in the full paper.

Lyrics are another important component. The system will provide a way for the user to be able to retrieve[8], view and edit the lyrics of a song.

Videos

Movies are an important part of each person's life. Everyone wants to have fast access to his or her movies. Linux provides several powerful *command line* programs for video playback. The purpose of our media center is to provide a GUI for these programs.

Media Assistant

Even if we did our best to automatically configure the system, some issues might still require user intervention. This a bit of a problem for users that have no computer knowledge. To overcome this, we have built a *media assistant*[9] function. This will display a cute character which will retrieve the messages provided by the auto configuration tool and will try to ask the user *simple* questions. This is doable, as mostly the questions that the user has to answer are rather simple. For instance, there might be a problem on detecting the video card type (NVIDIA or ATI). A user will usually know what card he has (he has paid a lot of money on it). With this simple information provided by the user, the system will be able to easily configure the graphic card. Another good example is the network configuration. If none of the *well-known* solutions work, a simple question like *Do you have a small paper with some pairs of four numbers separated by dots given to you by you Internet provider? Maybe a manual will help.* will solve the problem of the addresses.

Media Library

The file system is a real challenger for users. It is somehow hard to imagine folders into folders, different types of files that contain the same thing[10]. This is way our system tries to hide the file system from the user. This is done by means of a media library. A media scanner scans in background the whole

[7] The user is able to tell the system which songs to play next.

[8] Several online databases will be used (e.g. http://www.leoslyrics.com/)

[9] A similar approach was used by *Microsoft* in creating *Office Assistant*.

[10] Images for instance can be in several formats like JPEG, PNG, DIB etc. Music can be MP3, OGG, MOD etc. and videos have even more formats like AVI, MP4, DIVX etc.

hard drive and creates a database with multimedia content. In this way, the user will see a nice organized media library. It will not matter in which format a file is, the important information being the metadata of the file and the content.

This approach poses several challenges. First of all, files are located on the mobile media, on the host computer hard drives and on other portable devices. The necessity to move files around is a problem. Transferring files is still a problem that we are looking into. We have tested several solutions for this matter. The main issue was where to place the newly transferred file. For now, we use a special folder on each device where we place the files or a Windows Media Player similar solution[11].

Another problem posed by the media library is unifying the multiple metadata information. For instance, some music files contain some kind of metadata (sometime doubled – ID3 v1 and ID3 v2), others other type. We have tried to make the database as comprehensive as possible. Several results will be presented in the full paper.

Another important feature of the media library is the transfer of files to mobile media players. The user will be able to transfer files and synchronize to a mobile media player with just a click.

3. Technical Details

We will discuss now a few words about the technical details of the media center Linux.

Kernel

The kernel used for this system is 2.6. We have chosen this kernel due to its support for most of the generic multimedia devices. Most of the new hardware devices, like USB sticks, media players, sound cards, network cards and Intel WiFi cards, are directly supported by this kernel.

User Interface

The user interface is currently based on GTK+ 2 [Pe99], the same libraries as widely used GUIs[12] are based on. The design of the interface though is not done using entirely GTK Widgets (buttons, labels etc.), but also by using several custom graphics. The chosen platform for programming is C/C++ [Dr05].

All these libraries are stacked on top of the graphics Xorg server.

---

[11] Windows Media Player creates a folder structure similar to the media library one (Album \ Artist \ Song File).

[12] Gnome, Xfce

Multimedia Content

We have used several libraries in order to playback the media files. Music is mostly handled by FMOD Sound System, which delivers high performance. The only disadvantage is that its license is not free for commercial use. Still, it provides high performance as other free tools.

Videos are played back using MPlayer. This very flexible command line video player integrates itself very well in any GUI. It is perfect for our media center.

Media Library

The media library uses a database backend. For this, we have chosen two platforms, MySQL and Sqlite. We have tested which one suits better the system and came to the conclusion that both have ups and downs. Eventually, we have chosen the first option.

MySQL

MySQL has the advantage of being a fully featured transactional database system. The complexity, flexibility and speed of MySQL make it perfect for our system. As it works as a client-server application, the system might be extended to work over a network. This is useful if more media center systems where to be interconnected in a distributed system [TS07], thus a good starting point for improvement.

The drawback of this system is that it occupies a considerable space (more than 50 MB).

Sqlite

This is a very simple SQL library, which is linked into a program. It will store the database into a file (the system is similar to *FoxPro*). The great advantage is the size and complexity. It does not need more than a few KB for the library.

The drawback of this solution is that it cannot be used over a network. Moreover transactions and more complex queries are not supported. Still, it is a solution for our system's library.

Administration Tools

Besides the GUI, the system provides a professional administration tool for debugging and service reasons. These tools include software for experienced users, which will be able to fix a system which won't start with standard configuration.

Conclusions

The system that we have created is perfect suited for a home media center. Moreover, due to the fact that it is Linux based, it is very flexible and portable.

We have to underline that unlike other media center systems, which are in fact just some programs that run over a general purpose computer system, our solution is a stand alone system which is intended to be used only for the purpose of multimedia.

Most important, our system provides almost a complete auto configuration tool, meaning that the user does not have to know anything about computers. The system works exactly *like TV*, *out-of-the-box*. Moreover, it runs like a live system, requiring no installation. All that the user needs a CD, DVD or USB stick.

The user interface is designed by obeying the *three-clicks-away* rule, which states that anything should be done in no more than three clicks.

The file system is hidden from the user by means of a media library. This makes the usage of the system even simpler.

Besides the importance for the users, the features of this media center can be further used in normal Linux distribution. Auto configuration tools are features that are not present in almost any Linux system. This makes them hard to use for users with no computer knowledge.

Bibliography

[Be08] Beekmans, G; Linux From Scratch 6.4, http://www.linuxfromscratch.org/lfs/

[CM08] Centre Multimedia, http://fr.wikipedia.org/wiki/Media_center

[Dr05] Dragoi, G.; Programmation en C++ avec Visual C++ 6.0, Printech, Bucharest 2005

[FM08] Firelight Technologies, FMOD Wiki Documentation Page, http://www.fmod.org/wiki/index.php5?title=Main_Page

[GA06] Glass, G.; Ables, K.; Linux for Programmers and Users, Prentice Hall, 2006

[MT08] MPlayer Team, Mplayer – The Movie Player Documentation, http://www.mplayerhq.hu/DOCS/HTML/en/index.html

[Pe99] Pennington H.; GTK+/Gnome Application Development, New Riders Publishing, 1999

[So05] Sobell, M.; A Practical Guide to Linux Commands, Editors, and Shell Programming, Prentice Hall, 2005

[Ta04] Tannenbaum, A.; Sistemew de operare moderne, Editia a II-a (Modern Operating Systems, Second Edition), Byblos, Buchraest, 2004

[TS07] Tannenbaum, A; Steen M.; Distributed Systems – Principles and Paradigms, Pearson, 2007

[TW06] Tannebaum, A.; Woodhull, A, Operating Systems: Design and Implementation, Prentice Hall, 2006