

Zinc OS-Linux Based OS for Educational Institutes

¹Umesh Kumar, ²Dr. Neeta Sharma

¹Student, ²Head of Department

¹ Computer Science & Engineering,

¹Noida International University, Greater Noida, India

Abstract: Our world is changing rapidly because of marvels of technologies. This peculiar situation forces us to think about our education system. Earlier, books were perfect and only medium of knowledge and information, but world around is not same anymore. We need to introduce changes in education and teaching methodology for preparing a technologically literate generation.

As population grow, we need more education institutes to educate our next generations. These educational institutes need IT infrastructure to teach about the technologies and its practical implementation. We also need operating systems to cover the infrastructure with the technologies. Institutes need to invest more in windows operating system because of It's more user friendly. So We want to develop an operating system for educational institutes that can fulfill the complete requirement and can support the windows applications. So institutes can decrease the investment in windows operating system licensing [11].

We studied about the development of a common kernel that can support Linux and Windows system calls and provide a common platform for Linux packages and Windows softwares. Linux Kernel supports ELF[1] based executable formats but Windows NT Kernel supports PE[2] based executable formats. So it is very difficult to develop a common kernel and operating system which can support to Linux and Windows applications.

IndexTerms - Zinc OS, Wine Library, Linux Kernel, Ubuntu OS.

I. INTRODUCTION

As we all know that Windows operating system is widely used by the home users and students. Most of the educational institutes are using windows operating systems on systems to teach the students. Windows operating system is more popular because of its user friendly user interface and widely available software for the windows operating system.

Implementing the Linux operating system in educational institutes is good for the students as well for the institute in terms of finance because we all know that windows license is costly and Linux is open source and available free to use. If students will use Linux operating system in their college life, they can better understand the magic of code over the hardware, and a piece of code can do for them.

We are developing new technologies day by day and we are moving to the digital life very fast. So we have to provide an environment to the students where they can imagine the power of coding and hardware resource utilization using the code. Linux Operating system can provide this environment to the students easily

If we talk about the institutes, they need to purchase the costly licenses of windows operating system. Most of the organizations are not able to purchase these licenses and they are using cracked version of the windows operating system. It may cause to legal compliance [10] or Microsoft SAM review and audit process that leads to legal department for a full-blown Microsoft audit or legal case against your organization or huge fine charge on your organization.

We can solve all these issues with student's development for future technologies and saving lots of money by installing Linux operating system in institutes. I am agreeing that few software is available only for windows they don't have Linux version or Linux alternatives in the market. So we need to purchase only few copies of windows operating systems. And as we know that Linux is open source and we can customize [12] the Linux as per our requirements. So we are going to change the UI to make it more easy for students and other users. We are going to use few tools that can run windows based software on Linux operating system. So we can save the cost of windows licenses and we need to invest in software cost only.

We can also write an exokernel based operating system that can support Linux and Windows based applications but it is not that much easy to write an exokernel that can fulfil our requirements. Exokernel[3] have its own limitation like exokernel allocate the basic physical resources of the machine (e.g. disk blocks, memory pages, and processor time) to multiple application programs, and let each program decide what to do with these resources. The program can then link to a support library that implements the abstractions it needs (or it can implement its own).

FEATURES OF THE ZINC OS

A. User Interface

We are agreeing that linux have scary interface for windows friendly users but we also know that linux is an open source so we can change it as per our requirements. So we are going to change the UI of the linux operating system and make it more user friendly and looks like windows UI so that students can easily access the zinc operating system.

B. Software and Applications

We can install the any linux software that is available in the market of online stores on our zinc os. Mainly it will support the debian based software because of we are using ubuntu as our base operating system. But if user needs to install the rpm based softwares on zinc OS, we can install the additional package on zinc OS that can support the rpm packages on the zinc os.

Now these days we are more comfortable with apps because apps are easy to install and use. There is no need to manage the dependencies. So students and college can use most popular apps for linux from our app store. All the ubuntu store apps also available on our app store and user can use them.

C. Windows based software

We understand that there are many softwares are available for windows OS only. We are going to configure a compatibility layer in zinc OS to support the windows software in the zinc OS. It will reduce the cost of your windows licensing. But this compatibility layer may support the limited software and applications. And you can face issues related to the response of application.

. A compatibility layer [6] is an interface that allows binaries for a legacy or foreign system to run on a host system. This translates system calls for the foreign system into native system calls for the host system. With some libraries for the foreign system, this will often be sufficient to run foreign binaries on the host system. A hardware compatibility layer consists of tools that allow hardware emulation.

D. Custom Application

We will make this OS as per your requirements of the students. We can create multiple linux distros [7] for an educational institute as per their department requirements like Computer Department have more development tools, Electronics department have more electronics and electrical simulators and simple lab version have more browser and basic task utilities.

E. Security

We will allow to use the valid applications in zinc OS. And root user has all the permissions to install and remove the packages in the zinc OS. We are going to inherit all the security parameters of the ubuntu and try to customise them as per students and institutes.

F. Package manager and Repos

We are going to maintain our own package repositories to give you application in your single click. But you can also use the third party valid repos to install the new packages in zinc os.

Few institute have their own license of the applications and softwares so they can also manage their local reporsaties to install the application on their local network infrastructure based on zinc OS.

G. Updates

We are going to release the updates on the regular basis so that institutes can use zinc OS without any issues. We will maintain the online bug tracker so that user can send the bugs which they are facing and we will release the updates for all the users and institutes.

We are proving the custom OS to the college that means zinc OS will have the logo of the institute as the logo of OS and institute may also names their OS. So we have to manage the updates as below:

- Updates of bugs
- New Feature updates
- Compatibility layer updates
- Upgrade to higher versions

II. METHODOLOGY

As we know that Ubuntu is open source operating system and we can change the code as per our requirements. So changing the Ubuntu as per our requirements is not an impossible task for us. But when it comes to support .exe and .msi application on Linux. It is not possible to run any windows application on linux directly. So we are going to use a wine [5] library which supports the .net 3.0 and .net 3.5 based applications on linux.

We are going to use Ubuntu [4] 16.04 LTS edition as our base to develop the Zinc OS. We are going to remove all the unnecessary component from the Ubuntu OS and Install the necessary packages and libraries that are usable for the educational institutes. We are going to use the wine library to support the .exe and .msi application on our linux based Zinc OS.

The easiest way to create a custom operating system is to use Ubuntu Customization Kit. What follows is the manual [9] way of accomplishing the same. We are going to customize our operating system module wise so that we can test it on the devices that all the system calls are working perfectly fine.

A. Customization Environment

Before making changes in the base ubuntu operating system, we have to set up the chroot environment. We can make any necessary changes in the chroot environment in the ubuntu os.

We are going to use the following steps to make changes in ubuntu os:

B. Prepare and Chroot Environment

- Install the prerequisites to create chroot
sudo apt-get install squashfs-tools genisoimage
- Download the specific iso from the official website of ubuntu.
- Make a empty directory to extract the iso and make changes
mkdir ~/zincos
mv ubuntu-16.04-desktop-i386.iso ~/zincos
cd ~/zincos
- Now we are going to extract the .iso file
mkdir mnt
sudo mount -o loop ubuntu-16.04-desktop-i386.iso mnt
mkdir extract-iso
sudo rsync --exclude=/casper/filesystem.squashfs -a mnt/ extract-iso
- Now we have to extract the Desktop environment
sudo unsquashfs mnt/ casper/filesystem.squashfs
sudo mv squashfs-root edit
- Prepare the chroot. During this process we may lose the network connectivity inside the chroot environment so we have to make changes in the host files.
sudo cp /etc/resolv.conf edit/etc
sudo mount -o bind /run/ edit/run
sudo cp /etc/hosts edit/etc

Now we can prepare for the chroot

```
sudo mount --bind /dev/ edit/dev
sudo chroot edit
mount -t proc none /proc
mount -t sysfs none /sys
mount -t devpts none /dev/pts
```

To avoid the locate issues and in order to import GPG keys

```
export HOME=/root
export LC_ALL=C
```

C. Customizations

We are ready with the chroot environment. Now we can make changes in the packages and GNOME [8]

To manage the packages, we have to use the Apt package manager. But before installing and upgrading the packages you need to run:

```
dbus-uuidgen > /var/lib/dbus/machine-id
dpkg-divert --local --rename --add /sbin/initctl
ln -s /bin/true /sbin/initctl
```

Now your Apt is ready to install and remove the packages from the extracted iso. You can view all the installed packages by size:

```
dpkg-query -W --showformat='${Installed-Size}\t${Package}\n' | sort -nr | less
```

If you want to remove the package

```
aptitude purge package-name
```

If you want to install the package

```
apt-get install package-name (online repo)
```

```
dpkg -i package-install (from the .deb package)
```

D. Changing the UI

- We are going to change the backgrounds that inspire the students.
/usr/share/gnome-background-properties/ubuntu-wallpaper.xml
/usr/share/gconf/defaults/version_ubuntu-wallpapers
- Change gconf fonts, panels etc. We can edit the gconf-tree.xml file to make all the changes using gconf-2 gconftool-2 --direct --config-source.xml:readwrite:/etc/gconf/gconf.xml.defaults --type string --set yourkey "Value of Key"
- We can make it for legacy boot or based on UEFI boot process.
- Other than these changes we can change the language of the zinc operating system or keyboard to make it simple for the end users.

// In /usr/lib/ubiquity/ubiquity/misc.py line 620:

```
default_keymap = {
    'ar': 'ara',
    'bs': 'ba',
    'de': 'de',
    'el': 'gr',
    'en': 'us',
    etc...
```

E. Producing the Zinc CD Image

Before making the ISO of zinc OS we have to make number of changes in the extracted code and cleanup the environment so that our iso will be the bug free:

- Delete the user with UID 999 otherwise iso won't work.
- Set the defaults like language, keyboard layout, boot, locale settings
- If you want that your ISO will run the OS in live environment you can make changes for live environment and its permissions.
- Rebuild the initrd after making changes in the kernel
- Cleanup the environment
aptitude clean
- Delete the temporary and unnecessary files
rm -rf /tmp/* ~/.bash_history
rm /etc/hosts
rm /etc/resolv.conf
rm /var/lib/dbus/machine-id
rm /sbin/initctl
dpkg-divert --rename --remove /sbin/initctl
- Generate the ISO for zinc os
- Assembling the file system
chmod +w extract-cd/casper/filesystem.manifest
sudo su
chroot edit dpkg-query -W --showformat='\${Package} \${Version}\n' > extract-cd/casper/filesystem.manifest
exit
sudo cp extract-cd/casper/filesystem.manifest extract-cd/casper/filesystem.manifest-desktop
sudo sed -i '/ubiquity/d' extract-cd/casper/filesystem.manifest-desktop
sudo sed -i '/casper/d' extract-cd/casper/filesystem.manifest-desktop
- Compress filesystem
sudo rm extract-cd/casper/filesystem.squashfs
sudo mksquashfs edit extract-cd/casper/filesystem.squashfs -nolzma
- Set an image name in extract-cd/README.diskdefines
sudo vim extract-cd/README.diskdefines
- Remove old md5sum.txt and calculate new md5 sums
cd extract-cd
sudo rm md5sum.txt
find -type f -print0 | sudo xargs -0 md5sum | grep -v isolinux/boot.cat | sudo tee md5sum.txt
- Create the ISO image
sudo mkisofs -D -r -V "\$IMAGE_NAME" -cache-inodes -J -l -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o ../ubuntu-9.04.1-desktop-i386-custom.iso .

- Now we can test the ISO in our favorite virtual machine.

III. HARDWARE REQUIREMENTS

We don't need the higher configuration hardware to install our operating system. But when we are using the specific tools and applications on zinc OS then we need to improve the hardware as per the application requirements. So we are providing the specific hardware requirements that will fulfill the requirements of zinc OS and other application as well:

TABLE I. HARDWARE REQUIREMENTS

Sr No.	RESOURCE	REMARK
1.	PROCESSOR	DUAL CORE (1.2 GHZ MINIMUM)
2.	RAM	2 GB (MINIMUM)
3.	GPU	512 MB (MINIMUM)
4.	HDD	120 GB (MINIMUM)

IV. CHALLENGES

We are going to face many challenges in the implementation of Zinc OS in the institutes:

- Institutes may need the training of the linux operating system to implement it in their colleges.
- Institute may face that some of the windows application is not supported by zinc OS. We already discussed that wine compatibility layer does not support all the windows software. But we can reduce the institute investment in microsoft licensing.
- Institute needs some application or software for their general use on the zinc OS.
- Institute may need regular support to maintain the infrastructure and solve the issues in the real time.
- We have to maintain the stability and security of the zinc OS.

V. CONCLUSION

We are improving our technology day by day and we also need change the technology stack to support for the new technologies. If students will learn a full technology stack from their colleges then they can learn more and innovate more.

Most of the future technologies are supported by the linux. All the following domain of future technology give better performance on the linux operating so zinc OS will give the best environment to learn these future technologies:

- Artificial Intelligence
- Machine Learning
- Virtualization
- Cloud Development
- Mix Reality
- Virtual Reality
- Video Intelligence
- Deep Learning
- Internet of Things
- Embedded System Development
- Circuit and Device driver development

So Implementing the zinc OS will be the best solution for the institutes in terms of reducing the licensing cost and students learning about the technologies.

VI. FUTURE WORK

We can improve the compatibility layer so that it can support updated .net framework based applications. And reduce the response time of the windows application over the compatibility layer.

We can implement the number of software that required for institute with our Zinc OS to make our operating system more usable for the institutes. We can also develop the custom application [13] as per the requirements of the institutes.

REFERENCES

- [1] Wikipedia 2018. Comparison of operating system kernels retrieved from https://en.wikipedia.org/wiki/Comparison_of_operating_system_kernels#cite_note-12 on 15-March-2018
- [2] Wikipedia 2018. Ubuntu Operating system retrieved from [https://en.wikipedia.org/wiki/Ubuntu_\(operating_system\)](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) on 20-March-2018

- [3] Kaashoek, M. Frans; O'Toole, James Jr (1995). "Exokernel: An Operating System Architecture for Application-Level Resource Management
- [4] Wikipedia 2018. Ubuntu Operating system retrieved from [https://en.wikipedia.org/wiki/Ubuntu_\(operating_system\)](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) on 20-March-2018
- [5] Wikipedia 2018. Wine software for linux retrieved from [https://en.wikipedia.org/wiki/Wine_\(software\)](https://en.wikipedia.org/wiki/Wine_(software)) on 5- April- 2018
- [6] Wikipedia 2018. Windows compatibility layer retrieved from https://en.wikipedia.org/wiki/Compatibility_layer on 6-April-2018
- [7] Wikipedia 2018. Linux Distributions retrieved from https://en.wikipedia.org/wiki/Linux_distribution on 10-April-2018
- [8] Wikipedia 2018. GNOME UI for linux Operating system retrieved from <https://en.wikipedia.org/wiki/GNOME> on 10-April-2018
- [9] Wikipedia 2018 Live CD Customization of ubuntu retrieved from <https://help.ubuntu.com/community/LiveCDCustomization> on 12-April-2018
- [10] Rajala, R./Rossi, M. Tuunainen, V. K. 2003: A framework for analyzing software business models, Presented in the European Conference on Information Systems, Naples, Italy.
- [11] Windows Licensing. 2011. Windows Multipoint server 2012. Retrieved from <http://www.microsoft.com/windows/multipoint/buy.aspx> on 10 December 2013.
- [12] C. T. Lee, Z. W. Hong, and J. M. Lin, "Linux kernel customization for embedded system by using call graph approach," Asia and South Pacific Design Automation Conference, 2003, pp. 689-692.
- [13] C. T. Lee, Z. W. Hong, and J. M. Lin, "An application-oriented Linux kernel customization for embedded systems," National Computer Symposium, 2003.

