

USER'S

MANUAL

Joint Operating System

USER'S MANUAL
TABLE OF CONTENTS

1.0	GENERAL INFORMATION	2
1.1	System Overview.....	2
1.2	Authorized Use Permission	2
1.3	Points of Contact.....	2
1.3.1	Information	2
1.4	Organization of the Manual	3
1.5	Acronyms and Abbreviation.....	4
2.0	SYSTEM SUMMARY	6
2.1	System Configuration.....	6
2.2	User Access Level	6
2.3	Contingencies.....	6
3.0	GETTING STARTED.....	8
3.1	Environment Preparation	8
3.2	Dependencies Addition.....	8
3.2.1	GNU Compiler Collection	8
3.2.2	Quick Emulator.....	9
3.2.3	Pluma	10
3.2.4	Hexedit	10
3.3	JOS File Structure	11
3.3	Compiling and Building JOS.....	11
3.3	JOS Terminal.....	12
3.4	Exit System	12
4.0	USING THE SYSTEM	14
4.1	Command Line Interface.....	14
4.1.1	Clearing the Screen	15
4.1.2	Current Working Directory.....	15
4.1.3	Making a new Directory	15
4.1.4	Creating a File.....	15
4.1.5	Listing Directory Contents.....	15

4.1.6	Opening the File	16
4.1.7	Copying the File.....	16
4.1.8	Moving the File	16
4.1.9	Deleting File and Folder	17
4.1.10	Processes Dump.....	17
4.1.10	Killing Process.....	17
4.2	Graphical User Interface	18
4.2.1	Opening and Editing File	19

1.0 GENERAL INFORMATION

1.0 GENERAL INFORMATION

General Information section explains in general terms the Joint Operating System and the purpose for which it is intended.

1.1 System Overview

JOS is an operating system, which is an open source operating system with Swahili language feature for performing different operations such as files and folders management. JOS facilitate interaction and much understanding of how the operating system works because of using Swahili as the interaction language between the user and the JOS itself.

With that it makes the JOS to be used by every person widely in Tanzania and other countries who knows Swahili especially students who are in primary and secondary schools and make them understand the full operations of Operating System and IT skills knowledge in general. Currently the JOS operate on emulators such as QEMU which can be installed on the computer having Linux or Window platform.

1.2 Authorized Use Permission

JOS is an open source operating system, so it can be used by anyone for end user operations. Also its source codes are available which makes other developers to contribute and modify the JOS to make it to be more efficient.

Don't forget to acknowledge JOS developers when you modify and distribute the JOS source codes and to include the Wakuzaji.txt file having the name list of JOS developers.

1.3 Points of Contact

1.3.1 Information

If you encounter any problem or difficultness while building or using the JOS such as errors and bugs, please don't hesitate to communicate with us. Also if you have an idea or advice which you think it will improve the functions and operations of the JOS we are here to listen so share it with us.

Below is the list of JOS developers with their respective phone number;

Full name	Phone number
Abdallah, Mohamed	+255 718 551 278
Anthony, Timothy	+255 713 951 795
Giliad, Elifuraha M	+255 755 039 000
Onesphory, Eurad	+255 753 662 636
Richard, Elisha	+255 719 885 710
Tairo, Goodluck	+255 659 609 660

1.4 Organization of the Manual

The user's manual consists of four sections: General Information, System Summary, Getting Started and Using The System.

General Information section explains in general terms the Joint Operating System and the purpose for which it is intended.

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration and system's behavior in case of any contingencies.

Getting Started section explains how to get JOS, building and running it on the emulator such as QEMU. The section also presents briefly about JOS Terminal.

Using The System section provides a detailed description of system functions and how the user can interact with the JOS to get the desired outputs.

1.5 Acronyms and Abbreviation

Below is the list of all abbreviations and their meaning which are used in this user manual;

Abbreviations	Meaning
JOS	Joint Operating System
QEMU	Quick Emulator
IT	Information Technology
GCC	GNU Compiler Collection
MDI	Multiple Document Interface
RAM	Random Access Memory
CLI	Command Line Interface

2.0 SYSTEM SUMMARY

2.0 SYSTEM SUMMARY

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration and system's behavior in case of any contingencies.

2.1 System Configuration

JOS operates on emulators such as QEMU. The emulators can be used with Linux or Window platforms. JOS does not require connection to Internet for working, but you may need to have the internet connection to install and update the tools or dependencies which will be needed while building and running the JOS.

After the first clean building, the JOS will create a file system called `jos_fs.img` which will be used as the hard disk to store all the information and files of user and JOS itself. The file system help you to run the JOS later on without building it again, which can be done by using the file `washa.sh`

2.2 User Access Level

Everyone can use the Joint Operating System to perform various tasks, for example user can do files and folders management operations such as creating, viewing, removing, copying and moving different files and folders from one location to another.

2.3 Contingencies

All JOS information and user information such as files and folders are stored in the JOS file system called `jos_fs.img` which is used as the hard disk, this will help to maintain the information in case there is outage of power or un-intended closing of the Emulator which run JOS. But when you perform the new clean building of the JOS, it will replace the current file system and wipe the previous data. So make sure to copy or perform the backup of the file system if present before performing a new building of the Joint Operating System.

3.0 GETTING STARTED

3.0 GETTING STARTED

Getting Started section explains how to get JOS, building and running it on the emulator such as QEMU. The section also presents briefly about JOS Terminal.

3.1 Environment Preparation

Before being able to build and run the JOS, you need to prepare and configure the working environment. This include the computer running either a Linux or Window based Operating System having a minimum requirements of 1 GHz processor and 256 MB of RAM. You can use the computer having Window as operating system but we advise you to use the one running any Linux distro because it will be easy to install or update the tools and dependencies needed for building and running JOS. We tested the JOS in a computer running Ubuntu Mate 16 without any problem.

3.2 Dependencies Addition

Also you need to download and configure the tools and dependencies such as GCC, QEMU, Pluma and Hexedit which helps in the whole process of building and running JOS.

The commands used in this manual document for installation and configuration of different tools and dependencies were run on a computer running Ubuntu Mate. If you using another Linux distro and encounter any problem find the appropriate command to install and configure it, or find the appropriate tool or dependencies which will work on Windows in case you are using the computer which is installed with the Windows.

3.2.1 GNU Compiler Collection

The GNU Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain.

GCC will be used to compile Joint Operating System source codes files of .c and .asm into .o files which are object files. And then GCC will link those object files to create a different single files which will be in binary format and executable. In our case it will create a binary files for Bootloader and Kernel.

Mostly Linux distro comes with GCC already, in case you do not have it, you can use the following command(s) to install it. *(Type the command on your Terminal, and don't forget to supply the root password if asked)*

```
sudo apt-get install gcc
```

To verify if the GCC is already installed you can use one of the following commands;

```
gcc -v
```

```
gcc -version
```

If you still encounter any problem installing GCC, type the following commands one after another;
(This will install GCC version 4.8)

```
sudo apt-get install python-software-properties
```

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
```

```
sudo apt-get update
```

```
sudo apt-get install gcc-4.8
```

```
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 50
```

3.2.2 Quick Emulator

QEMU is a generic and open source machine emulator and virtualizer. It is free and open source tool that allows users to create and manage Virtual machines inside the host operating system. The resources of the host operating system, such as hard drive, RAM, Processor, will be divided and shared by the guest operating systems (Virtual machines).

QEMU will be used to run the Joint Operating System by taking the values of the JOS image file, file system as the parameters and the number of processor and amount of RAM to use.

To install QEMU, write the following command on your Terminal.

```
sudo apt-get install qemu-kvm qemu virt-manager virt-viewer  
libvirt-bin
```

You may use other Emulator such as Virtualbox and Bochs which they both offer the same functionalities.

3.2.3 Pluma

Pluma is a graphical application which supports editing multiple text files in one window (tabs or MDI). It fully supports international text through its use of the Unicode UTF-8 encoding. As a general purpose text editor, pluma supports most standard editor features, and emphasizes simplicity and ease of use.

Pluma will be used to view the source codes of the JOS, which will allow you to do more configuration and adding new functionalities by modifying the source codes.

Mostly pluma comes as the default text editor for the Ubuntu Mate distro, but you can use the following command to install it;

```
sudo apt-get install pluma-dev
```

If you encounter with any problem while installing, type the following commands one after another.

```
sudo apt-get update  
sudo apt-get install pluma-dev
```

3.2.4 Hexedit

Hexedit is among of the editors which deals with binary or byte files. Hexedit is a type of computer program that allows manipulation of the fundamental binary data that constitutes a computer file. The name 'hex' comes from 'hexadecimal': a standard numerical format for representing binary data.

Hexedit will be used to view the content of the linked executable binary file i.e. Bootloader and Kernel and verifying them if they are in good format of values and size.

To install Hexedit on your computer, type the following commands on your Terminal one after another;

```
sudo apt-get update  
sudo apt-get install hexedit
```

3.3 JOS File Structure

These are the most important files and directories in the Joint Operating System:

- **j os/** - Contains the entire Joint Operating System source codes.
- **j os/bitmaps/** - Contains the JOS background images and icons.
- **j os/boot/** - Contains the JOS bootloader source codes.
- **j os/components/** - Contains JOS kernel source codes.
- **j os/documentations/** - Contains JOS user manual and other document files.
- **j os/images/** - Contains JOS image files.
- **j os/objects/** - Contains JOS object files.
- **j os/programs** - Contains JOS user programs source codes.
- **j os/kijenzi.sh** - JOS building file.

3.3 Compiling and Building JOS

The process of compiling and building the Joint Operating is performed by following these steps:

- i. On your terminal navigate to where the folder of JOS source codes present.
- ii. Change the permission of all the files and folders of JOS source codes. You can use the following command on your terminal.

```
chmod 777 -R jos
```
- iii. Change the current working directory to be that of the JOS source code. You can use the following command on your terminal.

```
cd jos
```
- iv. Run the building file called kijenzi.sh present on the root of the JOS source codes files. You can use the following command on your terminal

```
./kijenzi.sh
```

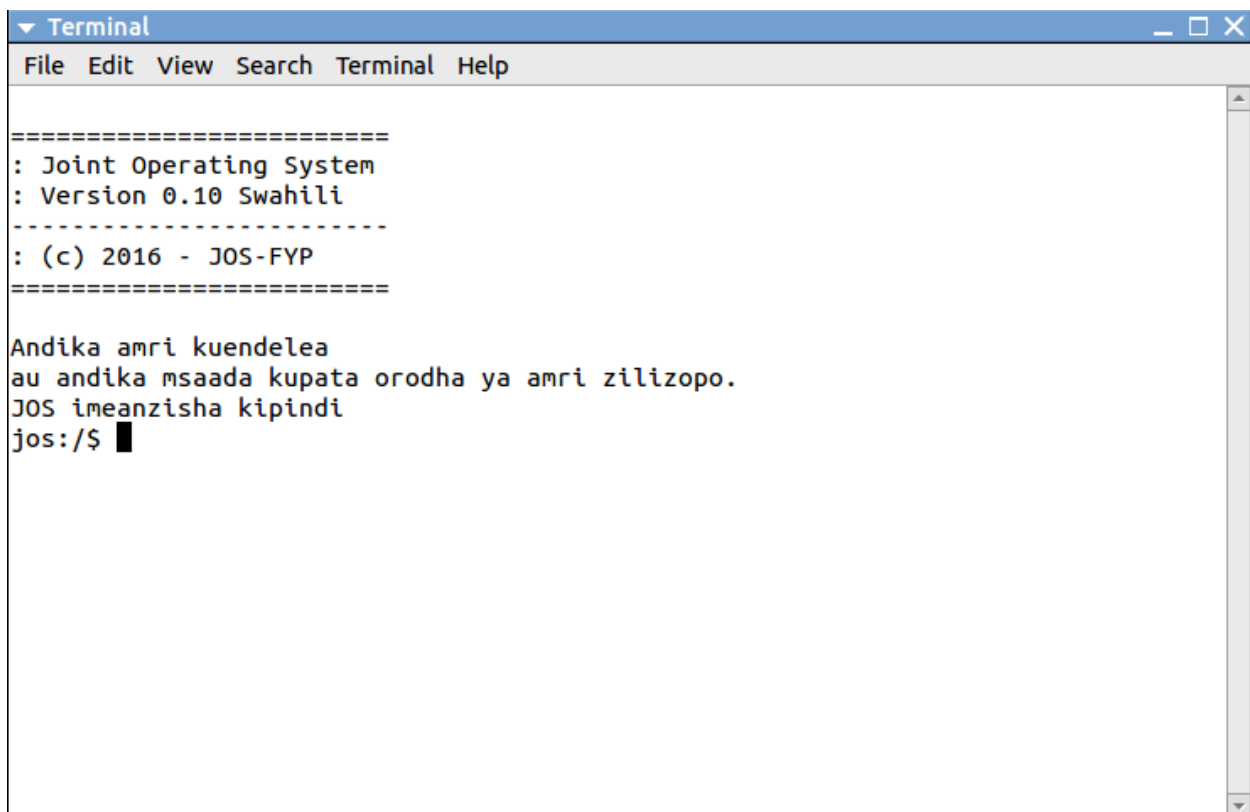
Or you can double click the kijenzi.sh file and choose to run it in terminal.

Then you should wait for the process of JOS compilation and building to be finished and be presented with the JOS Terminal as the first interface which is a CLI for the interaction between the user and the JOS itself.

3.3 JOS Terminal

After finishing the process of compiling and building the Joint Operating System and everything went well you will be given the JOS Terminal as the first interaction interface. If you cannot see the JOS Terminal or encounter any problem or difficulty while compiling and building the Joint Operating System, you should verify if you installed all required tools and dependencies, and you followed all the procedures of compiling and building the Joint Operating System.

You will be provided with the following JOS Terminal:



```
▼ Terminal
File Edit View Search Terminal Help

=====
: Joint Operating System
: Version 0.10 Swahili
-----
: (c) 2016 - JOS-FYP
=====

Andika amri kuendelea
au andika msaada kupata orodha ya amri zilizopo.
JOS imeanzisha kipindi
jos:/$ █
```

3.4 Exit System

Joint Operating System can be shut down and exit by entering the command “zima” on the JOS terminal, or since JOS runs on the emulator you can close the emulator which will close and exit both the emulator and the JOS itself too.

4.0 USING THE SYSTEM

4.0 USING THE SYSTEM

Using the System section provides a detailed description of system functions and how the user can interact with the JOS to get the desired outputs.

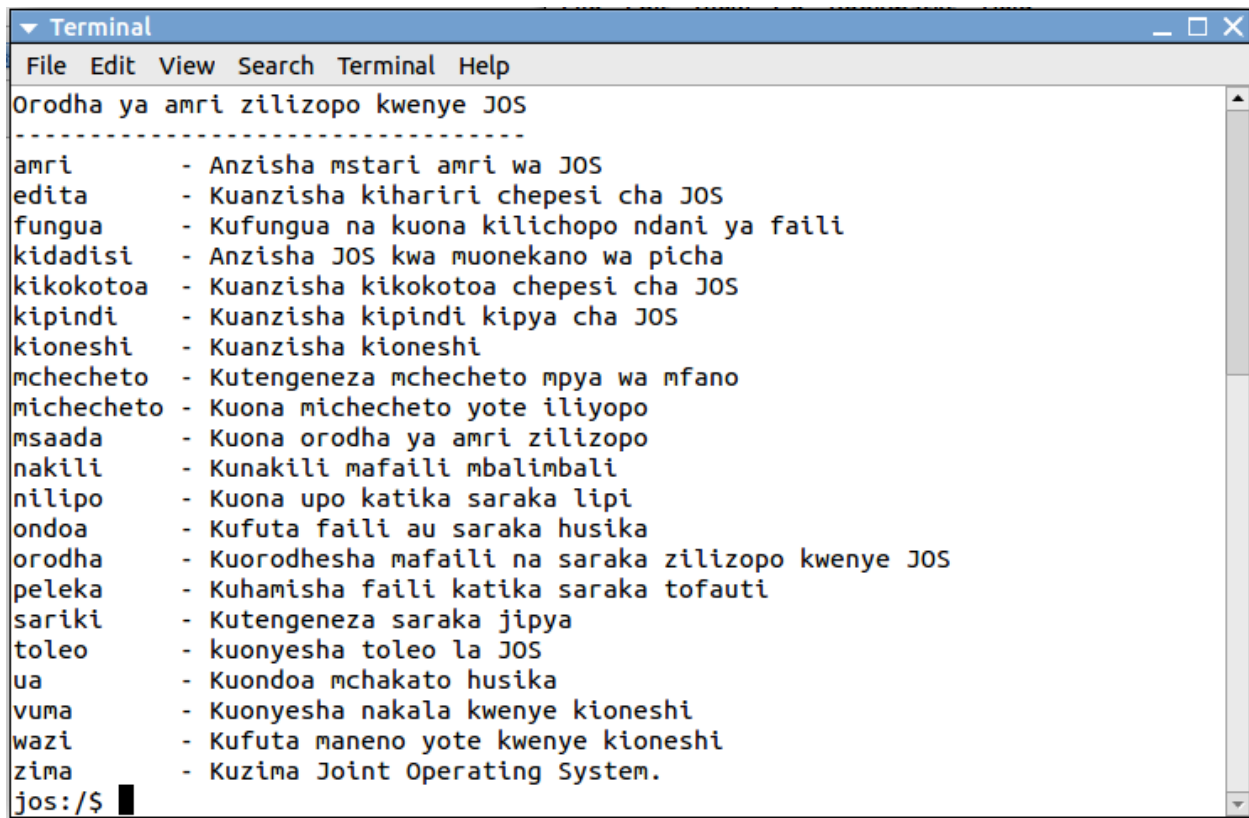
4.1 Command Line Interface

On the Command Line of Joint Operating System it's where you type your command to interact with the operating system kernel. You can start with "msaada" command which will give you the list of all available commands.

You can write that command like this;

```
jos:/ $ kidadisi
```

And you will be provided with the following on the terminal screen.



```
▼ Terminal
File Edit View Search Terminal Help
Orodha ya amri zilizopo kwenye JOS
-----
amri      - Anzisha mstari amri wa JOS
edita     - Kuanzisha kihariri chepesi cha JOS
fungua    - Kufungua na kuona kilichopo ndani ya faili
kidadisi  - Anzisha JOS kwa muonekano wa picha
kikokotoa - Kuanzisha kikokotoa chepesi cha JOS
kipindi   - Kuanzisha kipindi kipya cha JOS
kioneshi  - Kuanzisha kioneshi
mchecheto - Kutengeneza mchecheto mpya wa mfano
michecheto - Kuona michecheto yote iliyopo
msaada    - Kuona orodha ya amri zilizopo
nakili    - Kunakili mafaili mbalimbali
nilipo    - Kuona upo katika saraka lipi
ondoa     - Kufuta faili au saraka husika
orodha    - Kuorodhesha mafaili na saraka zilizopo kwenye JOS
peleka    - Kuhamisha faili katika saraka tofauti
sariki    - Kutengeneza saraka jipya
toleo     - kuonyesha toleo la JOS
ua        - Kuondoa mchakato husika
vuma      - Kuonyesha nakala kwenye kioneshi
wazi      - Kufuta maneno yote kwenye kioneshi
zima      - Kuzima Joint Operating System.
jos:/ $
```

From that you will be given with the list of all available commands and their meaning which you can use in the JOS to perform various specific tasks.

4.1.1 Clearing the Screen

To clear the screen of the JOS terminal you will need to write the command “wazi” on the terminal. The command can be written like this;

```
j os:/ $ wazi
```

4.1.2 Current Working Directory

To know the path and name of the directory you are current working on, you need to use the command “nilipo” on the JOS terminal. The command is written like this;

```
j os:/ $ nilipo
```

4.1.3 Making a new Directory

To create the new directory you need to use the command “sariki” followed with a folders names as the arguments, its syntax is *sariki foldername1 foldername2*

The command may be written like this;

```
j os:/ $ sariki folder1
j os:/ $ sariki folder2 folder3
j os:/ $ sariki folder1/folder1A folder2/folder2A folder4
```

4.1.4 Creating a File

To create the new file you need to use the command “edita”, which is a program for creating and editing files. The edita command needs to receive the name of the file as the argument, but if you wont supply it, it will create the file with the default name which is “faili.txt”

The command syntax is *edita filename.extension*, and it can be written like this;

```
j os:/ $ edita
j os:/ $ edita document1.txt
```

4.1.5 Listing Directory Contents

To list the contents of a certain folder or directory, you need to use the command “onesha”. The command will need to be passed with the path name of the directory or it will use the current working directoty.

The command may be used like this;

```
j os:/ $ onesha  
j os:/ $ onesha folder1  
j os:/ $ onesha folder1/folder1A
```

4.1.6 Opening the File

To open the file you need to use the command “fungua” on the JOS terminal. The command must be used with the filename as the argument. It’s syntax is *fungua filename*

The command may be used like this;

```
j os:/ $ fungua faili.txt  
j os:/ $ fungua folder1/document1.txt
```

4.1.7 Copying the File

To copy the file you need to use the command “nakili” on the JOS terminal. The command must be used with the two arguments, one for the source file path and another for the destination file path. And it must be specified with both source and destination file names.

It’s syntax is *nakili source_file_path destination_file_path*

The command may be used like this;

```
j os:/ $ nakili faili.txt faili_copy.txt  
j os:/ $ nakili folder1/doc1.txt folder2/folder2A/doc_copy.txt
```

4.1.8 Moving the File

To move the file you need to use the command “peleka” on the JOS terminal. The command must be used with the two arguments, one for the source file path and another for the destination file path. And it must be specified with source file name and not with the destination file name.

It’s syntax is *peleka source_file_path destination_file_path*

The command may be used like this;

```
j os:/ $ peleka faili.txt /  
j os:/ $ peleka folder1/document1.txt folder2/folder2A/
```

4.1.9 Deleting File and Folder

To remove the file you need to use the command “ondosha” on the JOS terminal. The command must be used with the file or folder source path.

It’s syntax is *ondosha source_file_or_folder_path*

The command may be used like this;

```
j os:/ $ ondosha faili.txt  
j os:/ $ ondosha folder1  
j os:/ $ ondosha folder2/folder2A
```

4.1.10 Processes Dump

To dump or list of all current working process, you need to use the command “michecheto” on the JOS terminal.

The command may be used like this;

```
j os:/ $ michecheto
```

4.1.10 Killing Process

To kill specific process which is unwanted you need to use the command “ua” on the JOS terminal. The command must be followed with the process id as the argument. It should be performed after dumping the current working process.

It’s syntax is *ua process_id*

The command may be used like this;

```
j os:/ $ ua 23
```

4.2 Graphical User Interface

On the JOS Terminal which offers Command Line Interface, you can switch between the CLI and GUI simply by writing “kidadisi” as the command name on the JOS Terminal.

```
j os:/ $ kidadisi
```



The QEMU will switch from CLI mode to GUI and presenting you with the view as shown above, which allows you to manipulate the files and folders easily.

4.2.1 Opening and Editing File

You can hover the mouse to the specific file you want and double click it to open, the EditPlus application of the JOS will open the file and you can view the content of the file.

When you are opening the file with EditPlus you can delete, add and edit the content of the file and save them by clicking the “Hifadhi” Button which will store the new content of the opened file.

