



POTATO OPERATING SYSTEM USERS GUIDE

Hasan Ibraheem, Henry Vos, Nicholas Fryer, Jay Kmetz

Preface

Welcome to the user's manual for the Potato Operating System! This manual will serve as a reference and guide for anyone to download, install, and use the Potato Operating System. We have documented every step that a user may need to operate the Potato Operating System. There is also a Help Section at the end of this manual. Thank you for using Potato Operating System!

Table of Contents

| | |
|---|-----------|
| PREFACE | 1 |
| DEFINITIONS AND ABBREVIATIONS | 3 |
| DOWNLOADING THE POTATO OPERATING SYSTEM..... | 4 |
| BUILDING THE POTATO OPERATING SYSTEM..... | 5 |
| LAUNCHING THE POTATO OPERATING SYSTEM | 6 |
| USING THE POTATO OPERATING SYSTEM | 7 |
| <i>Date.....</i> | <i>7</i> |
| <i>History</i> | <i>7</i> |
| <i>Help</i> | <i>8</i> |
| <i>Shutdown</i> | <i>8</i> |
| <i>Time.....</i> | <i>9</i> |
| <i>Version</i> | <i>10</i> |
| HOW TO GET HELP | 11 |

Definitions and Abbreviations

- Command – A command is an operation that performs a specific task as specified by the command's description. A user can request a command to be performed while using the POS
- Flags/Options/Parameters – A flag is special parameter that the user can append to a command to alter or modify the output of that command. Not every command has flags and not all flags operate the same
- Potato Operating System (POS) – The operating system that the user will be interacting with while using this product

Downloading the Potato Operating System

POS is a private operating system. If you would like to download and use the POS, you must request access to the developers of POS. You may find the developers contact information in the *How to Get Help* section. Please email one of the developers to get access to the POS.

Once you have confirmed your access with one of the developers, you may begin the download process for POS. The developers use the git protocol with the bitbucket git repository. Once you are ready, proceed to the bitbucket git repository and download the entire project to your local system. This is the process to download the POS.

Building the Potato Operating System

The POS is designed to be built before use. This is a very important step. If the POS is not built before attempting to run, it will not work at all. Below are the steps to build the POS.

1. Navigate to the root of the POS directory
2. Change directory to `pos_core`
3. Inside the `pos_core` there will be a file called `MakeFile`
4. To initiate the creation of POS, type *make* into your terminal environment

If there are any issues with the above steps, please reach out to a developer for assistance. You can find the developers contact information in the *How to Get Help* section.

Launching the Potato Operating System

To launch the POS, you must first build the POS. If you have not built the system yet, please refer to the section *Building the Potato Operating System*. Follow the instructions below to launch the POS.

1. Navigate to the root directory of the POS directory
2. Change directory to pos_core
3. Execute the following command inside your terminal environment
`qemu-system-i386 -nographic -kernel kernel.bin -s`

The POS will begin, and you should be at the POS command line input. At this point, POS is ready to begin standard operation and accept input from the user.

Using the Potato Operating System

Since the POS is a command line operating system, you will have to issue commands through your keyboard. All commands and their features will be keyboard driven for the POS. To get more information about the individual commands that can be run, please refer to each command inside the *Commands* section below.

Date

Description:

The user can query the system date and set the system date with the date command.

Options:

The date command has one parameter.

- [-s | --set] – To set the system date

Usage:

If date is used without any parameters, the current system date will be displayed. If you use the set date flag, -s or --set, it will attempt to set the system date to whatever date you have given to the command. The date format is MM:DD:YYYY with the following restrictions.

- All parameters are integers
- Months are between one and twelve
- Days are between one and thirty-one, depending on the month and year
- Years are between 1700 and 2999

Examples:

- To print out the current date
 - `date`
- To change the system date to February 12, 1997
 - `date -s 02/12/1997`
- To change the system date to December 28, 2870
 - `date --set 12/28/2870`

History

Description:

Built into the POS is a native history utility. Every command that you use will be logged into a system file. This enables the user to scroll through previous commands that they have executed. These commands are executable once they are encountered in the history queue. You can hit enter to execute the command.

Options:

There are no parameters for the history utility.

Usage:

Once at least one command has been executed, you can scroll back through all the commands that have been entered up to XXX commands. You can scroll through the stored commands using the up and down arrow keys.

Examples:

- Scroll up once one command has been executed, use the up-arrow key to see that command again.

Help

Description:

The user can use the help command to request a help page for any other command.

Options:

The help command has two parameters.

- [-c | --command] – To request a brief help file for a certain command
- [-f | --full] – To request the full help file for a certain command

Usage:

If you use help without any parameters, it will return a list of all the commands that you can request a help page for. You can request the help page for any command using the -c or –*command* flags. This will print out a brief explanation of the command. You can request a full explanation using the -f or --full flags.

Examples:

- To request the list of commands that you can view the help page for
 - *help*
- To request the brief help page for a certain command
 - *help --command version*
- To request the full help page for a certain command
 - *help -c time --full*

Shutdown

Description:

The user can request to shut down the POS in a safe manner by using the shutdown command. When the shutdown command is initiated, POS will ask the user for shutdown confirmation. The user must enter yes for POS to be shutdown.

Options:

There are currently no parameters for the shutdown flag.

Usage:

There are two key words that can be used to activate the shutdown command. You can either use *shutdown* or *exit* to initiate the shutdown command.

Examples:

- To shut down the POS using the *exit* keyword
 - *exit*
- To shut down the POS using the *shutdown* keyword
 - *shutdown*

Time

Description:

The user can query the system time and set the system time with the time command.

Options:

The time command has one parameter.

- `[-s | --set]` – To set the system time

Usage:

If time is used without any parameters, the current system time will be displayed. If you use the set time flag, `-s` or `--set`, it will attempt to set the system time to whatever time you have given to the command. The time format is HH:MM:SS with the following restrictions.

- All parameters must be two integers
- The clock is always in 24-hour time
- Hours are greater than zero and less than 24
- Minutes are greater than zero and less than 60
- Seconds are greater than zero and less than 60

Examples:

- To print out the current time
 - *date*
- To change the system time to noon
 - *date -s 12:00:00*
- To change the system time to 8:27:30 PM
 - *date --set 20:27:30*

Description:

The user can use the version command to request version information about the POS.

Options:

The version command has one parameter.

- [-f | --full] – To request the full version of the POS

Usage:

If you use version without any parameters, it will return shortened version information. You can request the full version information by using the using the *-f* or *--full* flags.

Examples:

- To request the shortened version information
 - *version*
- To request the full version information
 - *version --full*

How to Get Help

The developers of POS have mitigated as many errors as possible during the production of POS. If you encounter any errors or bugs, please reach out to one of us to report this found error. We also encourage all POS users to reach out to us for comments and suggestions. Below you will find the contact information for all the developers of POS.

Hasan Ibraheem
hai0003@mix.wvu.edu

Nicholas Fryer
ndfryer@mix.wvu.edu

Henry Vos
hlv0002@mix.wvu.edu

Jay Kmetz
jek0025@mix.wvu.edu