POTAT-OS OPERATING SYSTEM USERS GUIDE

Hasan Ibraheem, Henry Vos, Nicholas Fryer, Jay Kmetz

Preface

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

Table of Contents

PREFACE **1**

DEFINITIONS AND ABBREVIATIONS **3**

DOWNLOADING THE POTATOS **4**

BUILDING THE POTATOS **5**

LAUNCHING THE POTATOS **6**

COMMAND OPTION SYNTAX **7**

USING THE POTATOS **8**

Alarm *8*
Alias *8*
Date *9*
Delete PCB *9*
Help *10*
History *10*
Infinity *11*
Loadr3 ..... *11*
Potat *12*
Resume PCB *12*
Set PCB Priority *13*
Show All PCBs *13*
Show Blocked PCBs *13*
Show PCB *14*
Show Memory *14*
Show Allocated Memory *15*
Show Free Memory *15*
Show Ready PCBs *15*
Show Suspended Ready PCBs *16*
Shutdown *16*
Suspend PCB *17*
Time *17*
Version *18*

HOW TO GET HELP **19**

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 PotatOS
- 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 (PotatOS) – The operating system that the user will be interacting with while using this product

Downloading the PotatOS

PotatOS is a private operating system. If you would like to download and use the PotatOS, you must request access to the developers of PotatOS. 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 PotatOS.

Once you have confirmed your access with one of the developers, you may begin the download process for PotatOS. 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 PotatOS.

Building the PotatOS

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

1. Navigate to the root of the PotatOS directory
2. Change directory to mpx_core
3. Inside the mpx_core there will be a file called MakeFile
4. To initiate the creation of PotatOS, 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 PotatOS

To launch the PotatOS, you must first build the PotatOS. 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 PotatOS.

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

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

Command Option Syntax

In the following section we will be covering all of the commands and functionality that PotatOS has to offer. But before we start reviewing the power of PotatOS, we need to explain the syntax of the command options.

There are two type of markings that you may see when reading this documentation concerning arguments for PotatOS commands. They are “less than & greater than” quotations and square brackets quotations.

The “less than & greater than” is used to signify something that is necessary to the command operation. It looks as follows *<command name>*. If this syntax is used in a command, it MUST be used, or the command will fail. It is usually used to specify a name or some specific reference to the command.

The square brackets is used for optional arguments to a command. They have a multitude of meanings and they are independent per command. An argument *-a* for one command may not mean the same thing for another command. In some commands, if an optional command is not used, there is a default value that will be set. In others, that argument will just be skipped. Please fully read this document to understand how these arguments will work while using PotatOS.

Using the PotatOS

Since the PotatOS 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 PotatOS. To get more information about the induvial commands that can be run, please refer to each command inside the *Commands* section below.

Alarm

Description:
The user can make an alarm that will display a desired message at a given date and time.

Options:
The alias command has four arguments

- [-s | --set] – Set a new alarm
- [-d | --date] – The date and time you wish to set the alarm on. If you omit the date, your current day will be assumed
- [-r | --remove] – Remove an alarm by referencing its name
- [-l | --list] – List all the current alarms

Usage:
Alarm has 4 different options that can be used. The set option must always be used with the date option. If you do not include a date with the date option, it will assume your current date. But you must always provide a time.

Examples:

- To set an alarm
    - *alarm -s Test alarm -d 12/12/2020 12:00:00*
    - *alarm –set Wake up -d 16:00:00*

Alias

Description:
The user can alias a native command with any alias command they wish. Note that the alias command must not contain spaces.

Options:
The alias command has two arguments

- <native command> - The name of the native, already on system, command that you are trying to alias
- <alias command> - The name of the alias command that will be the new alias.

Usage:
Alias takes two arguments which are both necessary. The native command is the name of the command that the use is trying to alias. And the alias command is the name of the new alias for the native command.

Examples:

- To alias help as h
  - *alias help h*
- To alias version as v
  - *alias version v*

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*

Delete PCB

Description:
The user can delete a PCB using the *deletePCB* command.

Options:
The Delete PCB command has one argument

- <command name> - The name of the command that will be deleted

Usage:
Delete PCB will delete a PCB if that requested PCB does exist. If the PCB does not exist, Delete PCB will inform the user of the failure.

Examples:

- To delete the PCB named “test”
  - *deletePCB test*
- To delete the PCB name “4”
  - *deletePCB 4*

Help

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

Options:
The help command has one optional command.

- <command name> - If you want to view the help page of a specific command, you append the desired command name to the help 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 appending the desired command to the end of a new help command request. This will print out the help page for that specific command.

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 version*
- To request the full help page for a certain command
  - *help time*

History

Description:
Built into the PotatOS 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 que. You

can hit enter to execute the command. You can also issue the history command to see a list of command that have been issued.

Options:
You can issue the history command by using the keyword *history*

Usage:
Once at least one command has been executed, you can scroll back through all the commands that have been entered up to 10 commands. You can scroll through the stored commands using the up and down arrow keys. You can also see the history by typing the command *history.*

Examples:

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

Infinity

Description:
The user can initiate the infinity process using the infinity command. The infinite command runs forever with a low priority to demonstrate context switching.

Options:
The infinite command has no arguments.

Usage:
The infinite command can be invoked by issuing the command infinity

Examples:

- To start the infinite process
    - *infinity*

Loadr3

Description:
The user can load the R3 module by using the *loadr3* command. This command is useful for testing and demonstration because it creates processes and initiates them into the process queues.

Options:

The *loadr3* command has no arguments.

Usage:
The *loadr3* command can be invoked by issuing the command *loadr3*

Examples:

- To load the Module 3 processes
    - *loadr3*

Potat

Description:
A novelty command that will display the Happy Potato, known as HapPotat, to the users terminal.

Options:
The *potat* command has no arguments.

Usage:
The *potat* command can be invoked by issuing the command *potat*

Examples:

- To display HapPotat
    - *potat*

Resume PCB

Description:
The user can Resume a PCB using the *resumePCB* command.

Options:
The *resumePCB* command has one argument

- <command name> - The name of the command that will be resumed

Usage:
Resume PCB will resume a PCB if that requested PCB does exist. If the PCB does not exist, resume PCB will inform the user of the failure.

Examples:

- To resume the PCB named “yellow”
    - *resumePCB yellow*
- To resume the PCB name “bat”

- *resumePCB bat*

Set PCB Priority

Description:
The user can set a PCB's priority using the *setPriorityPCB* command.

Options:
The *setPriorityPCB* command has two arguments

- <command name> - The name of the command that will be resumed
- [-p | --priority] – The new priority for the PCB

Usage:
Set PCB Priority will set a PCB's priority to the new value if the requested PCB does exist. If the PCB does not exist, Set PCB Priority will inform the user of the failure.

Examples:

- To set the priority of the PCB named "frog" to the priority 90
  - *setPriorityPCB frog -p 90*
- To set the priority of the PCB name "olive" to the priority 4
  - *setPriorityPCB olive --priority 4*

Show All PCBs

Description:
The user can show all PCBs using the *showAllPCBs* command.

Options:
The *showAllPCBs* command has no arguments

Usage:
The *showAllPCBs* command will list all the PCB that are currently on the PotatOS.

Examples:

- To show all PCBs on the PotatOS
  - *showAllPCBs*

Show Blocked PCBs

Description:
The user can show all Blocked PCBs using the *showBlockedPCBs* command.

Options:
The *showBlockedPCBs* command has no arguments

Usage:
The *showBlockedPCBs* command will list all the Blocked PCB that are currently on the PotatOS.

Examples:

- To show all Blocked PCBs on the PotatOS
    - *showBlockedPCBs*

Show PCB

Description:
The user can show a single PCB using the *showPCB* command.

Options:
The *setPriorityPCB* command has one argument.

- <command name> - The name of the command that will be shown

Usage:
The *showPCB* command will show the requested PCB. If the requested PCB does not exist, *showPCB* will tell the user that it could not find the desired PCB.

Examples:

- To show the PCB named “bright”
    - *showPCB bright*
- To show the PCB name “printer”
    - *showPCB printer*

Show Memory

Description:
The user can show the state of memory using the *showMem* command. This will list all current processes and their memory allocation, as well as the amount of free memory.

Options:
The *showMem* command has no arguments

Usage:
The *showMem* command will show the current state of all the memory heap.

Examples:

- To show the current memory state
    - *showMem*

Show Allocated Memory

Description:
The user can show the state of the allocated memory using the *showAllocMem* command. This will list all current processes and their memory allocation.

Options:
The *showAllocMem* command has no arguments

Usage:
The *showAllocMem* command will show the current state of all the allocated memory.

Examples:

- To show the current allocated memory state
    - *showAllocMem*

Show Free Memory

Description:
The user can show the state of the free memory using the *showFreeMem* command. This will show the free memory that is not currently allocated.

Options:
The *showFreeMem* command has no arguments

Usage:
The *showFreeMem* command will show the amount of free memory left in the heap.

Examples:

- To show the current free memory state
    - *showFreeMem*

Show Ready PCBs

Description:
The user can show all Ready PCBs using the *showReadyPCBs* command.

Options:
The *showReadyPCBs* command has no arguments

Usage:
The *showReadyPCBs* command will list all the Ready PCBs that are currently on the PotatOS.

Examples:

- To show all Ready PCBs on the PotatOS
    - *showReadyPCBs*

Show Suspended Ready PCBs

Description:
The user can show all Suspended Ready PCBs using the *showSusReadyPCBs* command.

Options:
The *showSusReadyPCBs* command has no arguments

Usage:
The *showSusReadyPCBs* command will list all the Suspended Ready PCBs that are currently on the PotatOS.

Examples:

- To show all Suspended Ready PCBs on the PotatOS
    - *showSusReadyPCBs*

Shutdown

Description:
The user can request to shut down the PotatOS in a safe manner by using the shutdown command. When the shutdown command is initiated, PotatOS will ask the user for shutdown confirmation. The user must enter *yes* for PotatOS 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 PotatOS using the *exit* keyword
    - *exit*
- To shut down the PotatOS using the *shutdown* keyword
    - *shutdown*

Suspend PCB

Description:
The user can suspend a single PCB using the *suspendPCB* command.

Options:
The *suspendPCB* command has one argument.

- <command name> - The name of the command that will be suspended

Usage:
The *suspendPCB* command will suspend the requested PCB. If the requested PCB does not exist, *suspendPCB* will tell the user that it could not find the desired PCB.

Examples:

- To suspend the PCB named “watch”
  - *suspendPCB watch*
- To suspend the PCB name “user”
  - *suspendPCB user*

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*

Version

Description:
The user can use the version command to request version information about the PotatOS.

Options:
The version command has one parameter.

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

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

Hasan Ibraheem
hai0003@mix.wvu.edu

Nicholas Fryer
ndfryer@mix.wvu.edu

Henry Vos
hlv0002@mix.wvu.edu

Jay Kmetz
jek0025@mix.wvu.edu