**User’s Manual**

**Thunder Krackens**

**MODULE R5**

Jenn Nguyen

Haofan Zheng

Adam Chandler

Jafar Alkusaimi

4/7//2016

**Welcome to Thunder Kracken’s MPX**

MPX is an operating system created for academic purposes. Each member of the design team has taken careful steps to insure the quality of this operating system. We hope you enjoy using our operating system.

**Introduction**

As you begin to run MPX, you’ll be faced with a terminal. This terminal is the primary interface of MPX. Everything you do will be done through this terminal, so it is best to get familiar with it. MPX’s terminal functions much like other terminals you may be familiar with; to write in the terminal, use the keyboard to input letters. You can edit what you have already wrote in the terminal by navigating with the left and right arrow keys in their respective directions. To undo mistakes, the backspace button removes the character to the left of the cursor and the delete button removes the character that the cursor is currently highlighting. Once you are familiar with the input, you are ready to use MPX. Using MPX requires the ability to input commands to execute functions. These commands are listed below. Note that many commands begin with the subject they deal with. Functions beginning with “mpx” are general commands to be used in mpx. Functions beginning with “pcb” deal with process control blocks. Functions beginning with “mcb” deal with memory control blocks.

**Commands**

This section is comprised of different commands to be used in the terminal. Each command is formatted with its name followed by a colon and then it’s syntax. Note that square brackets indicate a variable clause.

alloc: mcb alloc [Size]

Allocates the given size of memory from the heap.

Arguments:

Size The size in bytes of the memory to be allocated

Exit Status: Returns success unless the size is invalid or there is not enough memory to allocate a memory block of that size.

alloc: mcb free [Index]

Frees memory at the given index from the heap.

Arguments:

Index The starting index of the memory block to be freed.

Exit Status: Returns success unless the index is not within the valid range.

init: mcb init [Size]

Initializes all of the memory available for the MPX in the form of a heap.

Arguments:

Size The size of memory in bytes

Exit Status: Succeeds unless size is invalid.

isempty: mcb isempty

Checks if the memory is empty or not.

Exit Status: Always Succeeds.

show: mcb show -allocated

Displays the address and size of allocated MCBs.

Exit Status: Always succeeds.

show: mcb show -free

Displays the address and size of allocated MCBs.

Exit Status: Always succeeds.

getdate: mpx getdate

Displays the current date to the user in format month/day/year.

Exit Status: Always succeeds.

gettime: mpx gettime

Displays the current time to the user in format hours:minutes:seconds.

Exit Status: Always succeeds.

help: help [command]

If used alone, displays usage instructions for every command or when used with an individual command, gives a detailed summary of how to use that command.

Arguments:

command The command to be described

Exit Status:

Returns success unless the given command is not a valid command.

LoadR3: mpx loadr3 [priority]

Allows the user to load all r3 processes into memory in the suspended ready state at the given priority.

Arguments:

priority The priority at which to load all of the r3 processes

Exit Status: Succeeds unless the given priority is not in the correct range of values.

setdate: mpx setdate [MM]/[DD]/[YYYY]

Allows the user to set the date to a given date.

Arguments:

MM Specified month must be two digits long.

DD Specified day must be two digits long.

YYYY Specified year must be four digits long.

Exit Status: Return success unless the given month, day, or year is an incorrect value.

settime: mpx settime [HH]:[MM]:[SS]

Allows the user to set the time to a given time.

Arguments:

HH Specified hours ranging from 00 (midnight) to 23 (1AM).

MM Specified minutes.

SS Specified seconds.

Exit Status: Return success unless the given time is invalid values.

shutdown: mpx -shutdown

Shuts down the operating system.

Exit Status: Always succeeds.

version: mpx version

Prints the current version of MPX and completion date.

Exit Status: Always succeeds.

block: pcb block [processName]

Puts the given process into the blocked state.

Attributes:

processName The name of the process to be blocked

Exit Status: Return success unless the given name does not apply to any process.

create: pcb create [processName] [processClass] [processPriority]

Creates a pcb with the given attributes

Attributes:

processName The name of the process to be created.

processClass The class of the process to be created (“system” or “application”).

processPriority The priority of the process to be created (ranging from 0-9).

Exit Status: Return success unless the given name, class, or priority are invalid.

resume: pcb resume [processName]

Puts the given process into the unsuspended state.

Arguments:

processName The name of the process to be unsuspended.

Exit Status: Return Success unless the given name does not apply to any process.

setpriority: pcb setpriority [processName] [processPriority]

Allows the user to change the priority of a process.

Arguments:

processName The name of the process

processPriority The new priority value (Must be a digit 0 – 9)

Exit Status Return success unless there is no process by the given name, priority is not a digit 0-9, or the process already has the given priority.

show -all: pcb show -all

Shows all processes and their attributes.

Exit Status: Always succeeds.

show -blocked: pcb show -blocked

Shows all blocked processes and their attributes.

Exit Status: Always succeeds.

show -name: pcb show -name [processName]

Shows all of the attributes of a given PCB

Attributes:

processName The name of the process to be shown.

Exit Status: Returns success unless there is no process by the given name.

show -ready: pcb show -ready

Shows all ready processes and their attributes.

Exit Status: Always succeeds.

suspend: pcb suspend [processName]

Puts the given process into the suspended state.

Arguments:

processName The name of the process to be suspended.

Exit Status: Return Success unless the given name does not apply to any process

unblock: pcb unblock [processName]

Puts the given process into the unblocked state.

Attributes:

processName The name of the process to be unblocked

Exit Status: Return success unless the given name does not apply to any process.