DiamondDogs

Operating System

CS 450 MPX

User's Manual

Version 6.0

Dec 7th 2021

Dakota Lacy, Jacob Taylor, Easton Thewes, Leah Cave
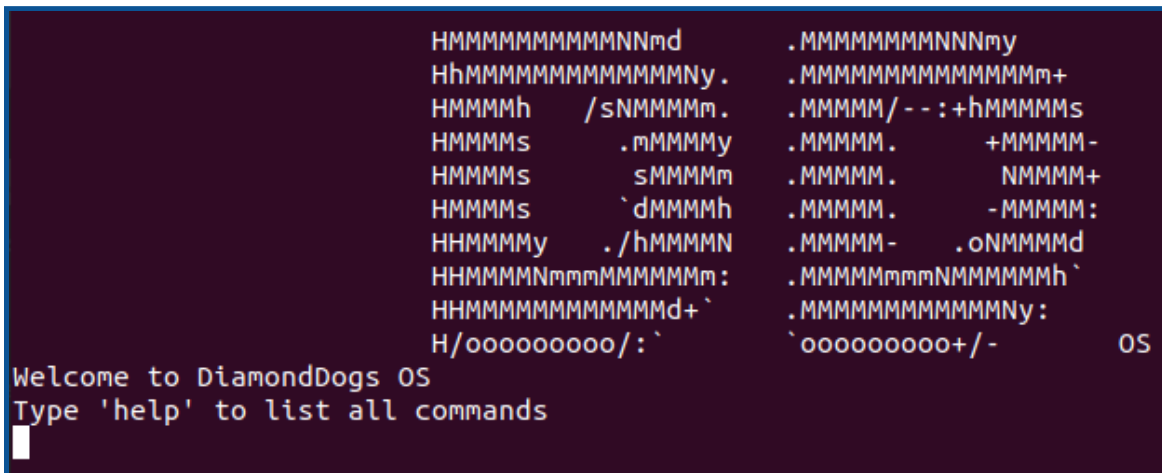
Contents

I. MPX Project Overview:

The MPX Project is designed to implement various functions of an operating system. This module includes our first version of the command menu, allowing the user to interact with the OS through the terminal. The user will use the keyboard to send input and whenever a system response is generated, it will display on the terminal window.

II. User Manual:

i. Getting Started

The initial boot of our MPX system will provide the user with details including the system name and instructions for how to view the menu. This menu will list each command the user has access to as well as a short description of its functionality. At the close of the menu, there is an additional feature that will specify how the user should implement the individual commands.

Figure 1: MPX DiamondDogs startup output.

```
                         HMMMMMMMMMMNNmd        .MMMMMMMMNNNmy
                         HhMMMMMMMMMMMMMMNy.    .MMMMMMMMMMMMMMMm+
                         HMMMMh    /sNMMMMm.    .MMMMM/--:+hMMMMMs
                         HMMMMs      .mMMMMy    .MMMMM.      +MMMMM-
                         HMMMMs        sMMMMm   .MMMMM.        NMMMM+
                         HMMMMs        `dMMMMh  .MMMMM.       -MMMMM:
                         HHMMMMy    ./hMMMMN    .MMMMM-    .oNMMMMd
                         HHMMMMNmmmMMMMMMMm:    .MMMMMmmmNMMMMMMh`
                         HHMMMMMMMMMMMMMMd+`    .MMMMMMMMMMMMMMNy:
                         H/ooooooooo/:`         `ooooooooo+/-       OS
Welcome to DiamondDogs OS
Type 'help' to list all commands
help


List of Commands:
'version' - To receive the OS current version.
'getdate' - To get the current date.
'gettime' - To get the current time.
'settime' - To set the current time.
'setdate' - To set the current date.
'shutdown' - To shutdown the OS.
'clear' - To clear the terminal window.
'suspendPCB' - to suspend selected PCB
'resumePCB' - to resume selected PCB
'setPriority' - to set new priority of selected PCB
'showPCB' - to show the contents of selected PCB
'showAll' - to show all PCB's in a queue
'showReady' - to show ready processes
'showBlocked' - to show blocked processes
'createPCB' - to create a PCB
'deletePCB' - to delete the selected PCB
'blockPCB' - to block a PCB
'unblockPCB' - to unblock a PCB

Please use <help 'commandname'> to view commands functionality.
```

Figure 2: User input highlighted with system response below.

ii.   General Commands

There are 8 general features included in the Version 2 menu that are directly accessible to the user. The options vary from simple data output to commands that directly update the OS based on user input. This section explains each command in more detail and provides an example of how to run each option.
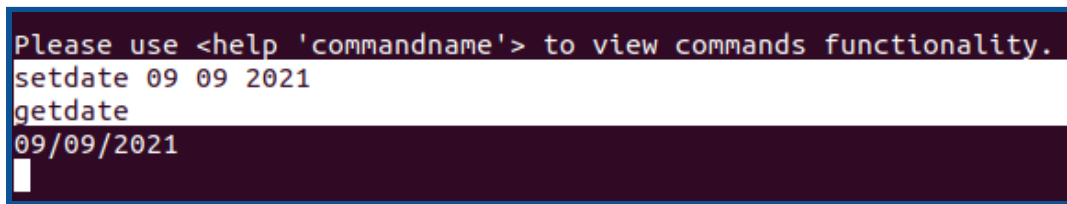
1. Get Date:

The get date command will output the date in MM/DD/YYYY format.

   $ **getdate**

2. Set Date:

The set date command will allow the user to specify the date, immediately updating the system. Input format should be HH MM SS as follows:

   $ **setdate 09 09 2021**


```
Please use <help 'commandname'> to view commands functionality.
setdate 09 09 2021
getdate
09/09/2021
```

Figure 3: User input, highlighted, showing the setdate & getdate commands.

3. Set Time:

The set time command will allow the user to enter in the time, automatically updating the OS.

   $ **settime 01 22 35**

4. Get Time:

The view time command outputs the current time in HH:MM:SS.

   $ **gettime**

```
Please use <help 'commandname'> to view commands functionality.
settime 01 22 34
gettime
01:22:36
```

Figure 4: The settime & gettime command syntax with execution response.

5. Version:

The version command will print out the operating system's current MPX version.

$ **version**

```
version

DiamondDogs OS Version R2.
```

Figure 5: Version command execution.

6. Help:

The help command provides a more in depth explanation corresponding to each menu option.

$ **help setdate**

```
Please use <help 'commandname'> to view commands functionality.
help setdate
Type 'setdate MM DD YYYY' to set the date to whatever date is current.
```

Figure 6: Help command execution.

7. Clear:

The clear command will clear the terminal window of all previous responses and commands without exiting the module. Cursor begins at the top left of the new window.

$ **clear**

```
                        HMMMMMMMMMMNNmd          .MMMMMMMMNNNmy
                        HhMMMMMMMMMMMMMMNy.       .MMMMMMMMMMMMMMMMm+
                        HMMMMh     /sNMMMMm.      .MMMMM/--:+hMMMMMs
                        HMMMMs       .mMMMMy      .MMMMM.      +MMMMM-
                        HMMMMs        sMMMMm      .MMMMM.       NMMMM+
                        HMMMMs       `dMMMMh      .MMMMM.      -MMMMM:
                        HHMMMMy    ./hMMMMN       .MMMMM-    .oNMMMMd
                        HHMMMMNmmmMMMMMMMm:       .MMMMMmmmNMMMMMMh`
                        HHMMMMMMMMMMMMMd+`        .MMMMMMMMMMMMMNy:
                        H/ooooooooo/:`           `ooooooooo+/-        OS
Welcome to DiamondDogs OS
Type 'help' to list all commands
clear
```

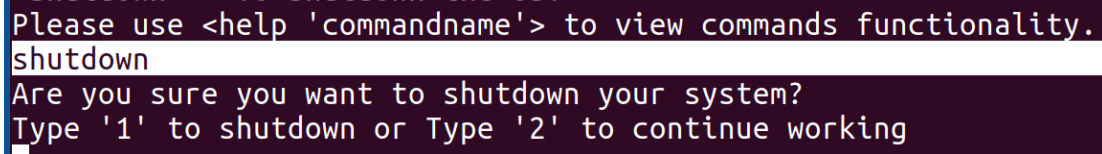*Figure 7: Operating system **prior** to executing the clear command.*

```

```

*Figure 8: Operating system **after** executing the clear command.*

8. Shutdown:

The shutdown command will first prompt the user to confirm they would like to exit the module and once the user confirms, the system proceeds to shut down. Any other response will exit the shutdown execution.
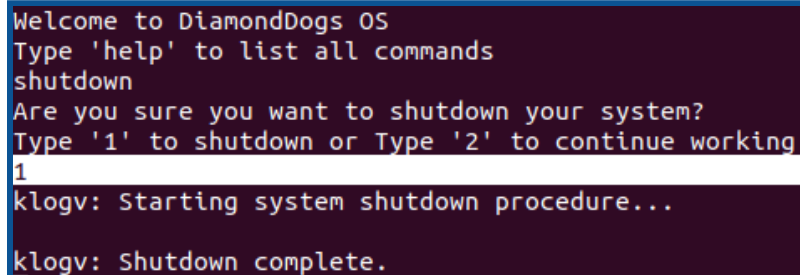
$ **shutdown**

```
Please use <help 'commandname'> to view commands functionality.
shutdown
Are you sure you want to shutdown your system?
Type '1' to shutdown or Type '2' to continue working
```
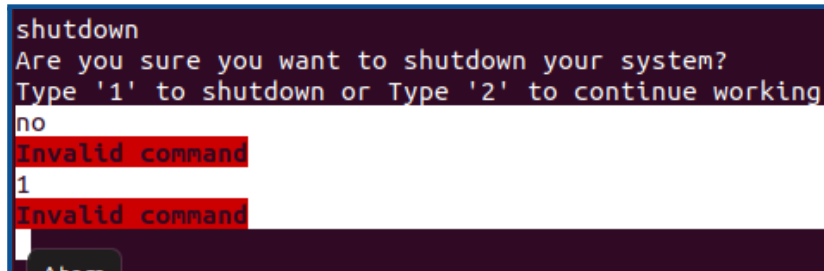
Figure 9: shutdown sequence ending with the prompt for user confirmation.

$ **1**

```
Welcome to DiamondDogs OS
Type 'help' to list all commands
shutdown
Are you sure you want to shutdown your system?
Type '1' to shutdown or Type '2' to continue working
1
klogv: Starting system shutdown procedure...

klogv: Shutdown complete.
```

Figure 10: shutdown command with user confirmation.

```
shutdown
Are you sure you want to shutdown your system?
Type '1' to shutdown or Type '2' to continue working
no
Invalid command
1
Invalid command
```

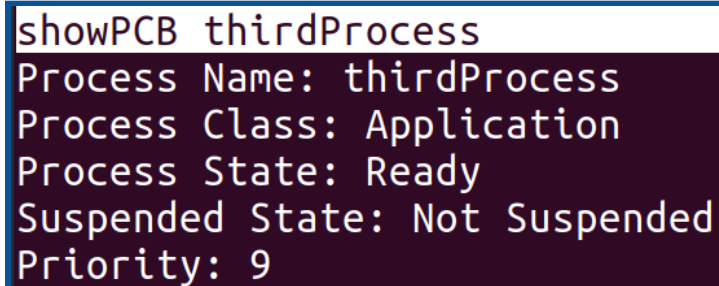*Figure 11: shutdown sequence terminating with invalid response.*

iii. Process Control Commands

There are 11 functions within Version 2 that relate specifically to the *process control block*, allowing the user to create/view/edit processes. This section explains each command in more detail and provides an example of how to run each option.

9. Show PCB:

ShowPCB displays specific process information including the Name, Class, State and Priority.

$ **showPCB <name>**

```
showPCB thirdProcess
Process Name: thirdProcess
Process Class: Application
Process State: Ready
Suspended State: Not Suspended
Priority: 9
```

Figure 13: Demonstrating showing a specific PCB

10. Show All:

Show all displays all PCB information for the processes in the ready queue as well as the blocked queue.

$ **showAll**

```
createPCB firstProcess 1 8
createPCB secondProcess 2 7
createPCB thirdProcess 2 9
showAll
Ready Queue:
- - - - - - - - - - -
Process Name: thirdProcess
Process Class: Application
Process State: Ready
Suspended State: Not Suspended
Priority: 9

- - - - - - - - - - -
Process Name: firstProcess
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 8

- - - - - - - - - - -
Process Name: secondProcess
Process Class: Application
Process State: Ready
Suspended State: Not Suspended
Priority: 7

Suspended Ready Queue:
Blocked Queue:
Suspended Blocked Queue:
```

Figure 15: Show all lists any and all current processes and designated queue.

11. Suspend PCB:

The suspend function will move the process into the suspended state and update its corresponding queue.

$ **suspendPCB <name>**

```
suspendPCB thirdProcess

Suspended Ready Queue:
- - - - - - - - - - - -
Process Name: thirdProcess
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 9

Blocked Queue:
Suspended Blocked Queue:
```

Figure 16: Suspend function immediately updating a processes state and queue.

12. Set Priority:

The setpriority is used to define a process's priority level and adjust it's position in the queue based on the difference.

$ **setPriority <name> <priority #>**

```
setPriority thirdProcess 1
priority set.
showPCB thirdProcess
Process Name: thirdProcess
Process Class: Application
Process State: Ready
Suspended State: Not Suspended
Priority: 1
```

Figure 15: Set priority command immediately updating a processes priority value.

13. Resume PCB:

Resume will allow the process to continue and update to its corresponding queue.

$ **resumePCB <name>**

Figure 17: Demonstrating a process being put back into the ready queue with 'resume' function.

14. Show Ready:

Show ready retrieves each block of information for each process in the Ready queue.

$ **showReady**

```
showReady
- - - - - - - - - - -
Process Name: secondProcess
Process Class: Application
Process State: Ready
Suspended State: Not Suspended
Priority: 7


- - - - - - - - - - -
Process Name: firstProcess
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 4

Suspended Ready Queue:
- - - - - - - - - - -
Process Name: thirdProcess
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 9
```

Figure 18: Show ready function displaying both the ready queue as well as the suspended ready queue.

15. Delete PCB:

deletePCB takes in a process name and once it is found, will remove it from the queue, link the remaining elements together and then free the space that had been previously allocated to that node.

$ **deletePCB <name>**

```
deletePCB firstProcess
Removed firstProcess
showAll
- - - - - - - - - - -
Process Name: secondProcess
Process Class: Application
Process State: Ready
Suspended State: Not Suspended
Priority: 7

Suspended Ready Queue:
- - - - - - - - - - -
Process Name: thirdProcess
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 9

Blocked Queue:
Suspended Blocked Queue:
```

Figure 19: The delete PCB function confirms a process has been removed and showAll displays the updated data.

16. Show Blocked:

Show blocked retrieves each block of information for each process in the Blocked queue.

$ **showBlocked**

```
blockPCB secondProcess
Process is blocked.
showBlocked
Blocked Queue:
-----------
Process Name: secondProcess
Process Class: Application
Process State: Blocked
Suspended State: Not Suspended
Priority: 7

Suspended Blocked Queue:
```

Figure 21: Show blocked displays both the blocked queue as well as the suspended blocked queue.

17. Alarm:

Creates an alarm structure to be added to the system, containing a message and time specified by the user.

$ **alarm <alarmMessage> <HH:MM:SS>**

```
alarm Example alarm message 09:20:55

Alarm1 has been added

Alarm1: Example alarm message has been dispatched
```

Figure 23: Alarm will create an alarm instance with specified data

18. Infinite:

Allows users to create an infinite process that will ultimately run until a user interrupt.

$ infinite

```
infinite
Infinite PROCESS EXECUTING.
showAll
Ready Queue:
------------
Process Name: INFINITE
Process Class: Application
Process State: Ready
Suspended State: Not Suspended
Priority: 5

Process Name: IDLE
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 0
------------
```

Figure 24: Alarm will create an alarm instance with specified data

19. Resume All:

This function changes all processes currently blocked or suspended to a ready state, initializing execution.

$ **resumeAll**

```
resumeAll
proc5 dispatched
showAll
Ready Queue:

Process Name: proc5
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 5

Process Name: proc4
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 4

Process Name: proc3
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 3

Process Name: proc2
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 2

Process Name: proc1
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 1

Process Name: IDLE
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 0
-----------
Suspended Ready Queue:
Blocked Queue:
Suspended Blocked Queue:
proc5 dispatched
```

Figure 25: Resume All dispatches 'proc5' upon user command and moves all processes to ready state.

20. Load R3:

Loads given processes into the system in a suspended ready state.

$ **loadr3**


```
loadr3

IDLE PROCESS EXECUTING.
showAll
Ready Queue:
-----------
Process Name: IDLE
Process Class: System
Process State: Ready
Suspended State: Not Suspended
Priority: 0
-----------
Suspended Ready Queue:

Process Name: proc5
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 5

Process Name: proc4
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 4

Process Name: proc3
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 3

Process Name: proc2
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 2

Process Name: proc1
Process Class: System
Process State: Ready
Suspended State: Suspended
Priority: 1
Blocked Queue:
Suspended Blocked Queue:
IDLE PROCESS EXECUTING.
```

Figure 26: Load R3 processes loaded and displayed using showAll.

21. Show Allocated:

Shows the allocated memory blocks inside of the user created heap

$**showAlloc**

```
showAlloc
Address: 218103828
Type: Allocated
Size: 1076

Address: 218104924
Type: Allocated
Size: 1076

Address: 218106020
Type: Allocated
Size: 1076

Address: 218107116
Type: Allocated
Size: 1076

Address: 218108212
Type: Allocated
Size: 1076

Address: 218109308
Type: Allocated
Size: 1076

Address: 218110404
Type: Allocated
Size: 1076
```

Figure 27: showAlloc with allocated memory blocks

22. Show Free:

Shows the free memory blocks inside of the user created Heap

```
showFree
Address: 218106020
Type: Free
Size: 47808
```

Figure 28: showFree with free memory blocks

23. init r6 function

This function allows the user to load the associated R6 inputs.