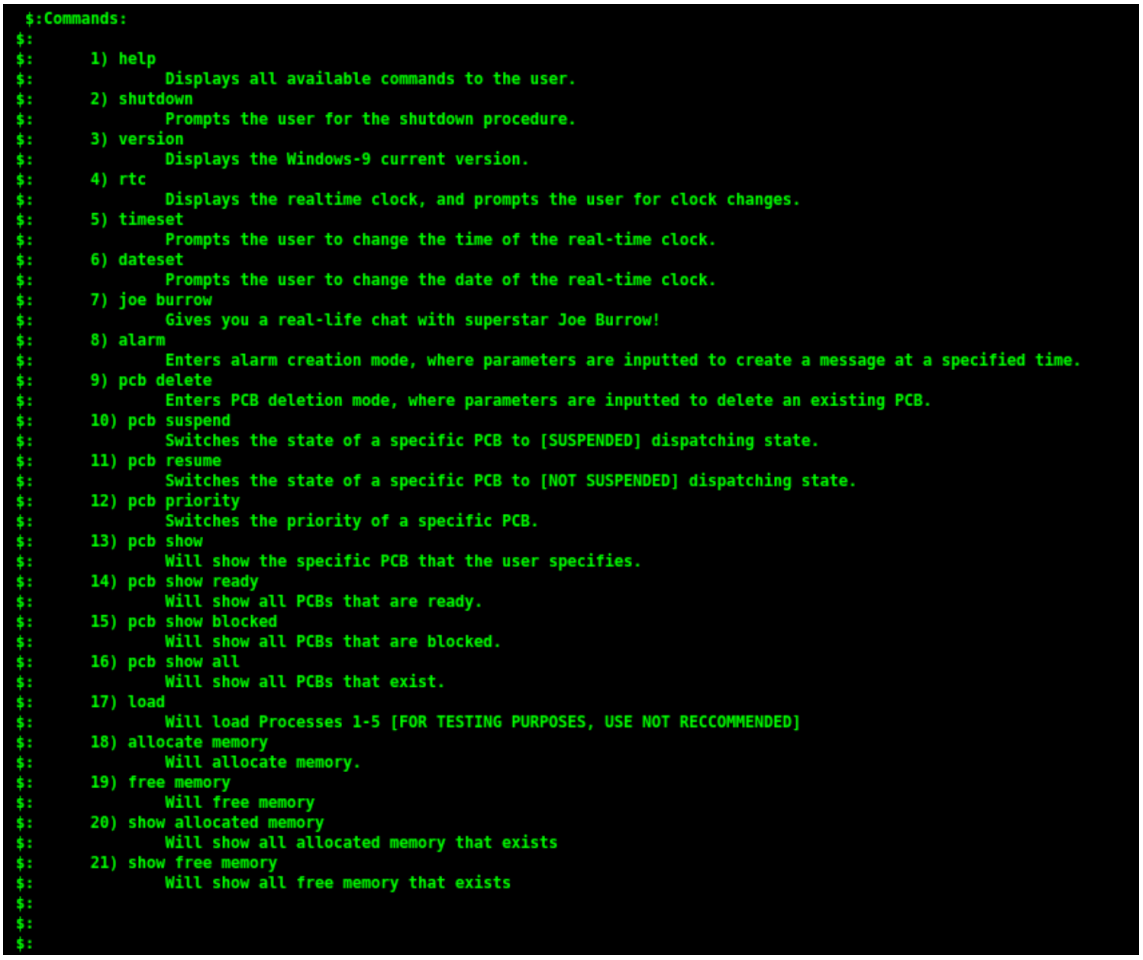
Commands

help

- ➢ **Command Name**
  - help
- ➢ **Command Description**
  - Displays a list of commands to the user. These commands, when inputted to the Operating System, will execute that specified process.
- ➢ **Command Example:**



shutdown

- ➢ **Command Name**
  - shutdown
- ➢ **Command Description**
  - Will prompt the user for shutdown. If the user specifies a ‘yes’, then the shutdown process starts. If not, shutdown will be aborted.
- ➢ **Command Example:**

```
> shutdown
shutdown
$:Are you sure you want to shutdown?:
$:      yes
$:      no

>yes
yesklogv: Starting system shutdown procedure...
klogv: Shutdown complete.
```

rtc

- ➢ **Command Name**
  - real time clock
- ➢ **Command Description**
  - Will display the user with the time and date from the internal real time clock, as specified through the *timeset* and *dataset* commands.
- ➢ **Command Example:**

```
> rtc
rtc
$:Real-Time Clock:
$:Current Time: 01:11:03
$:Current Date:02/03/23

>
```

timeset

- ➢ **Command Name**
  - set time
- ➢ **Command Description**
  - Will update the real time clock with the specified user input. Has to be inputted in a formatted way *(e.g 15 1/2 minutes past 12 noon can be represented as 12:15:30*)
- ➢ **Command Example:**

```
> timeset
timeset
$:Please enter a new time in the following format:
$:      HH:MM:SS
$:
$:      e.g [Fifteen and a half minutes past noon = 12:15:30]:

>12:15:30
12:15:30
$:Is this the time you'd like to set?
$: 12:15:30
$:      yes
$:      no

>yes
yes
$:Time has been changed to:
$:12:15:30
$:Returning to Menu...:
```

dateset

- ➢ **Command Name**
  - set date
- ➢ **Command Description**
  - Will update the real time clock with the specified user input. Has to be inputted in a formatted way *(e.g May 3rd, 2003 can be represented as 05/03/03*)
- ➢ **Command Example:**

```
> dateset
dateset
$:Please enter a new date in the following format:
$:      MM/DD/YY
$:
$:      e.g [February 18, 2008 = 02/18/08]:

>02/18/08
02/18/08
$:Is this the date you'd like to set?
$:  02/18/08
$:      yes
$:      no

  >yes
yes
$:Date has been changed to:
$:02/18/08
$:Returning to Menu...:
```

version

- ➢ **Command Name**
  - version
- ➢ **Command Description**
  - Will display the current version of the Operating System and compilation date to the user.
- ➢ **Command Example:**

pcb delete

- ➢ **Command Name**
  - pcb delete
- ➢ **Command Description**
  - pcb delete takes name of the pcb to delete, then will delete it if the pcb exists
- ➢ **Command Example**

```
> pcb delete
pcb delete
$:Please enter the name of the PCB you would like to delete:
> example
example
$:PCB deleted:
$:Returning to menu...:
```

pcb suspend

- ➢ **Command Name**
  - pcb suspend
- ➢ **Command Description**
  - takes the name of the pcb to suspend and switches it to the suspend dispatching state
- ➢ **Command Example**

```
> pcb suspend
pcb suspend
$:Please enter the name of the PCB you would like to switch to the [SUSPENDED] dispatching state:
> example
example
$:PCB example has been given the [SUSPENDED] dispatching state
```

pcb resume

- ➢ **Command Name**
  - pcb resume
- ➢ **Command Description**

- takes the name of the pcb to resume and switches it to the not suspended dispatching state

➢ **Command Example**

```
> pcb resume
pcb resume
$:Please enter the name of the PCB you would like to switch to the [NOT SUSPENDED] dispatching state:
> example
example
$:PCB example has been given the [NOT SUSPENDED] dispatching state
```

pcb priority

➢ **Command Name**

- Pcb priority

➢ **Command Description**

- takes name of the pcb to change the priority of, then will take the priority it is to be changed to. If the PCB exists and the priority is a valid value it changes the priority of the PCB to the one entered.

➢ **Command Example**

```
> pcb priority
pcb priority
$:Please enter the name of the PCB you would like to change priority:
> ex
ex
$:PCB ex currently has priority 3:
$:Please enter the new desired priority of PCB ex:
$:This number must range from [0-9].
> 2
2
$:PCB ex's priority set to 2 :
$:Returning to menu...
```

pcb show

➢ **Command Name**

- Pcb show

➢ **Command Description**

- takes name of the pcb to show, then will show it if the pcb exists.

➢ **Command Example**

```
> pcb show
pcb show
$:Please enter the name of the PCB you would like to show:
> ex
ex

$:PCB Name: ex
$:Priority: 2
$:Class Level: USER
$:Execution State: READY
$:Dispatching State: NOT_SUSPENDED

$:Returning to menu...
```

pcb show ready

- **Command Name**
  - Pcb show ready
- **Command Description**
  - Shows all the pcbs that are ready.
- **Command Example**

```
> pcb show ready
pcb show ready

$:PCB Name: a
$:Priority: 3
$:Class Level: USER
$:Execution State: READY
$:Dispatching State: NOT_SUSPENDED

$:Returning to menu...
```

pcb show blocked

- **Command Name**
  - Pcb show blocked
- **Command Description**
  - Shows all the pcbs that are blocked.
- **Command Example**

```
> pcb show blocked
pcb show blocked

$:PCB Name: a
$:Priority: 3
$:Class Level: USER
$:Execution State: BLOCKED
$:Dispatching State: NOT_SUSPENDED

$:Returning to menu...
```

pcb show all

- ➢ **Command Name**
  - Pcb show all
- ➢ **Command Description**
  - Shows all the pcbs.
- ➢ **Command Example**

```
> pcb show all
pcb show all

$:PCB Name: example1
$:Priority: 1
$:Class Level: USER
$:Execution State: READY
$:Dispatching State: NOT_SUSPENDED


$:PCB Name: example2
$:Priority: 2
$:Class Level: ADMIN
$:Execution State: READY
$:Dispatching State: NOT_SUSPENDED

$:All PCBs are shown above:
$:If you see no PCBs, no PCBs currently exist.
```

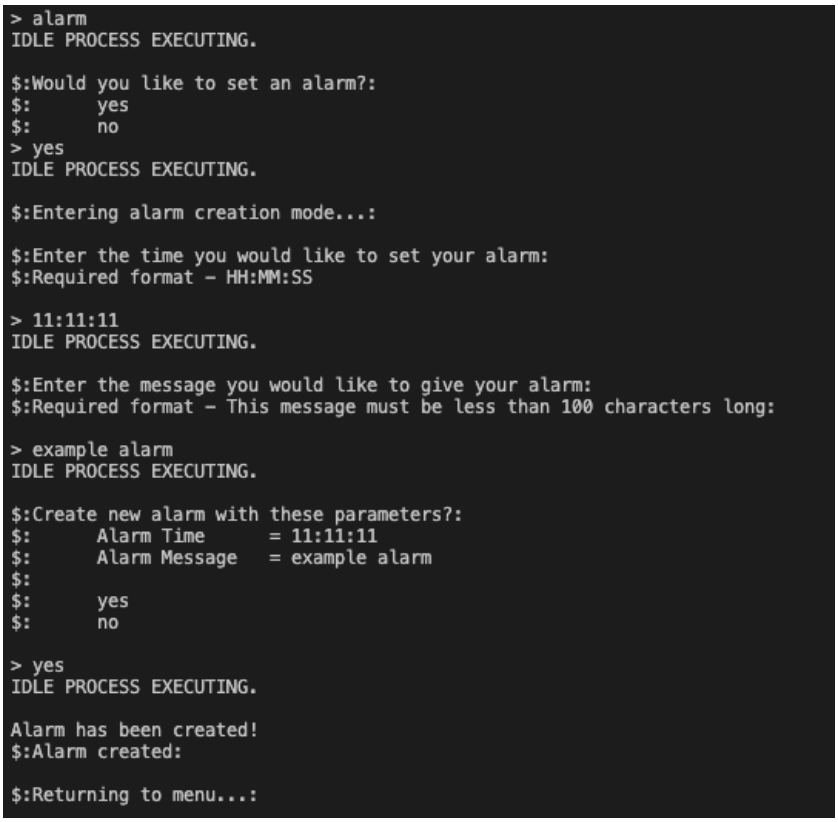
load

- ➢ **Command Name**
  - load
- ➢ **Command Description**
  - Loads the R3 test processes into a non-suspended ready state and initializes and saves the contexts for each process
- ➢ **Command Example**

```
> load
IDLE PROCESS EXECUTING.
```

alarm

- ➢ **Command Name**
  - alarm
- ➢ **Command Description**
  - Allows you to set an alarm at a certain time that will display a message when the alarm is triggered
- ➢ **Command Example**



```
> alarm
IDLE PROCESS EXECUTING.

$:Would you like to set an alarm?:
$:      yes
$:      no
> yes
IDLE PROCESS EXECUTING.

$:Entering alarm creation mode...:

$:Enter the time you would like to set your alarm:
$:Required format - HH:MM:SS

> 11:11:11
IDLE PROCESS EXECUTING.

$:Enter the message you would like to give your alarm:
$:Required format - This message must be less than 100 characters long:

> example alarm
IDLE PROCESS EXECUTING.

$:Create new alarm with these parameters?:
$:      Alarm Time      = 11:11:11
$:      Alarm Message   = example alarm
$:
$:      yes
$:      no

> yes
IDLE PROCESS EXECUTING.

Alarm has been created!
$:Alarm created:

$:Returning to menu...:
```

allocate memory

- ➢ **Command Name**
  - allocate memory
- ➢ **Command Description**
  - Will prompt the user for the size of the block to be allocated. If the user enters a valid value it allocates the memory of the specified size and displays the address. If not, an error message will appear and no memory will be allocated.
- ➢ **Command Example**

```
> allocate memory
IDLE PROCESS EXECUTING.

$:How much memory would you like to allocate?
$:Format: NUMERICAL VALUE WITH MAX LENGTH OF 15

> 21

$:Memory allocated to: 0xD000F50
$:Returning to menu...
```

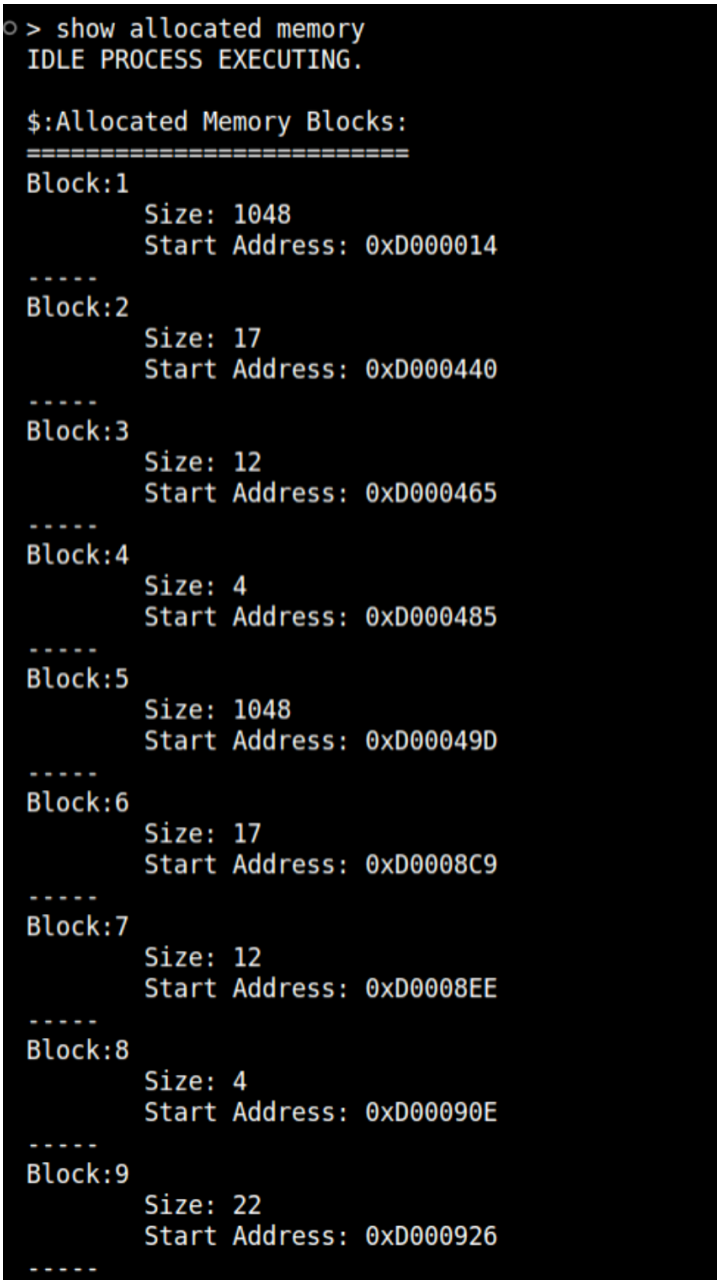
free memory

- ➢ **Command Name**
  - free memory
- ➢ **Command Description**
  - Will prompt the user for the start address of the block to be freed. If the user enters a valid value it frees the memory at the specified address. If not, an error message will appear and no memory will be freed.
- ➢ **Command Example**

```
> free memory
IDLE PROCESS EXECUTING.

$:Whats the address of the memory you would like to free?:
$:Format: HEXIDECIMAL VALUE WITH MAX LENGTH OF 15

> 0xD000F50

$:Memory block has been successfully freed!:

$:Returning to menu...
```

show allocated memory

- **Command Name**
  - show allocated memory
- **Command Description**
  - Shows all the allocated memory blocks with their block number, size, and address.
- **Command Example**



show free memory

- **Command Name**
  - show free memory
- **Command Description**
  - Shows all the free memory blocks with their block number, size, and address.
- **Command Example**

```
> show free memory
IDLE PROCESS EXECUTING.

$:Free Memory Blocks:
==========================
Block:1
        Size: 47407
        Start Address: 0xD000A35
==========================
$:Memory Blocks shown above:

$:If you see no Memory Blocks, none currently exist:
```