



West Virginia University®

BENJAMIN M. STATLER COLLEGE OF  
ENGINEERING AND MINERAL RESOURCES

# MarauderOS User's Manual

CS 450: Operating Systems Structure

Author: *Ian Jackson*

Instructor: *Dr. Camille Hayhurst*  
Department: Lane Department of CS and EE  
Date: October 2, 2023

---

# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Standard Commands</b>                    | <b>1</b> |
| 1.1      | version . . . . .                           | 1        |
| 1.2      | shutdown . . . . .                          | 1        |
| <b>2</b> | <b>Date &amp; Time Commands</b>             | <b>2</b> |
| 2.1      | getdate . . . . .                           | 2        |
| 2.2      | setdate . . . . .                           | 3        |
| 2.3      | gettime . . . . .                           | 4        |
| 2.4      | settime . . . . .                           | 4        |
| <b>3</b> | <b>Process Control Block (PCB) Commands</b> | <b>5</b> |
| 3.1      | Create a PCB . . . . .                      | 6        |
| 3.2      | Delete a PCB . . . . .                      | 7        |
| 3.3      | Block a PCB . . . . .                       | 7        |
| 3.4      | Unblock a PCB . . . . .                     | 8        |
| 3.5      | Suspend a PCB . . . . .                     | 9        |
| 3.6      | Resume a PCB . . . . .                      | 9        |
| 3.7      | Change a PCB's Priority . . . . .           | 10       |
| 3.8      | List Current PCBs . . . . .                 | 11       |
| 3.9      | Summary of PCB Commands . . . . .           | 13       |

# 1 Standard Commands

## 1.1 version

The `version` command in MarauderOS serves as a handy tool for users seeking to gain insights into the current state of the operating system. When executed, this command displays the version on the command line, including the OS's major and minor version numbers, along with the date it was compiled. The major version number signifies the milestone the OS is on. The minor version number signifies the number of features and smaller enhancements that have been incorporated since the last major release.

### Usage

The `version` command takes in no parameters. An example usage of the command can be seen below in Figure 1.



```

MarauderOS

Welcome to MarauderOS | Use 'help' command to see list of commands
> version
v1.9 last compiled on 09-08-23
> █
```

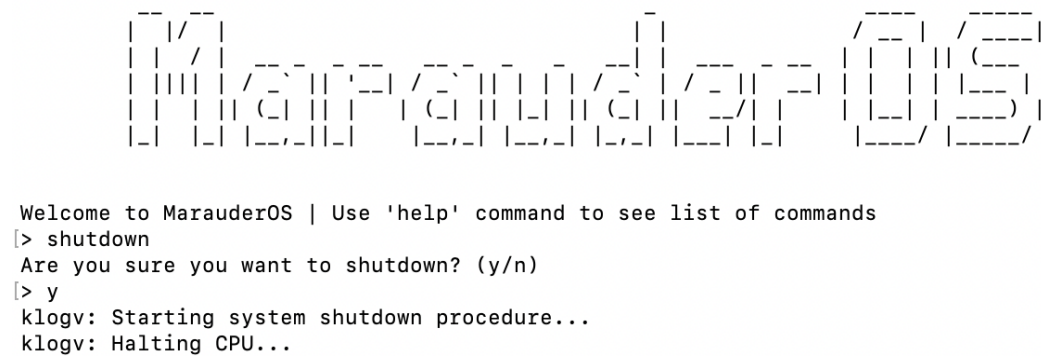
Figure 1: Example Usage of the `version` Command

## 1.2 shutdown

The `shutdown` command within MarauderOS serves as a way to halt the operating system's operations. When this command is executed, it asks the user if they are sure they want to shutdown the operating system. Its primary function is to bring the OS to a complete stop, ensuring a clean and efficient termination of all running processes, thus maintaining system stability and reliability. A user can immediately shutdown the system without confirmation by executing `shutdown!` instead.

## Usage

The `shutdown` command takes in no parameters. An example usage of the command can be seen below in Figure 2.



```
MarauderOS | Use 'help' command to see list of commands
[> shutdown
Are you sure you want to shutdown? (y/n)
[> y
klogv: Starting system shutdown procedure...
klogv: Halting CPU...
```

Figure 2: Example Usage of the `shutdown` Command

## 2 Date & Time Commands

### 2.1 `getdate`

The `getdate` command in MarauderOS provides users with a convenient way to retrieve and display the current date directly within the terminal interface. Upon execution, this command promptly generates the date in a user-friendly format: `mm-dd-yy`, where `mm` represents the numerical month, `dd` stands for the day of the month, and `yy` indicates the last two digits of the current year. The year is assumed to be in the 21st century, so the current years supported is from 2000 to 2099. The `getdate` command ensures that users can easily access and utilize the current date within the MarauderOS environment.

## Usage

The `getdate` command takes in no parameters. An example usage of the command can be seen below in Figure 3.



```
Welcome to MarauderOS | Use 'help' command to see list of commands
[> getdate
09-10-23
> █
```

Figure 3: Example Usage of the `getdate` Command

## 2.2 `setdate`

The `setdate` command in MarauderOS provides users with a practical means to configure and set the system's date directly from the terminal interface. This command accepts a single parameter, the desired date, provided in the format `mm-dd-yy`, where "mm" represents the numerical month, "dd" denotes the day of the month, and "yy" indicates the last two digits of the year. The `setdate` command empowers users to maintain precise control over the system's date configuration, enhancing the overall functionality and usability of MarauderOS.

### Usage

The `setdate` command takes in one parameter; the date the user wants to set the system to in the `mm-dd-yy` format. The user must use the `-d` flag before the date is specified. The user will face error messages if the date is out of range or format is incorrect. An example usage of the command can be seen below in Figure 4.



```
Welcome to MarauderOS | Use 'help' command to see list of commands
[> setdate -d 06-29-07
> █
```

Figure 4: Example Usage of the `setdate` Command

## 2.3 `gettime`

The `gettime` command in MarauderOS offers users a straightforward way to access and display the current time within the terminal interface. Upon execution, this command promptly provides the current time in Greenwich Mean Time (GMT), utilizing the 24-hour time format for enhanced precision. The time is presented in the format `hh:mm:ss`, where "hh" represents the hour, "mm" denotes the minutes, and "ss" indicates the seconds. The `gettime` command ensures efficient access to GMT in the 24-hour format within the MarauderOS environment.

### Usage

The `gettime` command takes in no parameters. An example usage of the command can be seen below in Figure 5.



```
MarauderOS - Terminal
Welcome to MarauderOS | Use 'help' command to see list of commands
[> gettime
13:55:21
> █
```

Figure 5: Example Usage of the `gettime` Command

## 2.4 `settime`

The `settime` command in MarauderOS provides users with a powerful tool to configure and set the system's time directly from the terminal interface. This command accepts a single parameter, which is the desired time in the 24-hour GMT format. The time is specified using the `hh:mm:ss` format, where "hh" represents the hour, "mm" denotes the minutes, and "ss" indicates the seconds. This command empowers users to maintain fine-grained control over the system's time configuration within MarauderOS, enhancing overall accuracy and usability.

## Usage

The `settime` command takes in one parameter; the time the user wants to set the system to in the `hh:mm:ss` format. The user must use the `-t` flag before the time is specified. The user will face error messages if the time is out of range or the format is incorrect. Note that the time is in 24-hour GMT format. An example usage of the command can be seen below in Figure 6.

The image shows a terminal window with a dark background. At the top, there is a large, stylized ASCII art logo that reads "MarauderOS". Below the logo, a welcome message says "Welcome to MarauderOS | Use 'help' command to see list of commands". The user has entered the command `> settime -t 09:41:00`, and the prompt has moved to the next line, indicating the command was executed successfully.

```

Welcome to MarauderOS | Use 'help' command to see list of commands
> settime -t 09:41:00
>
```

Figure 6: Example Usage of the `settime` Command

## 3 Process Control Block (PCB) Commands

The Process Control Block (PCB) commands are all under the `pcb` command. The PCB command can complete many actions: create a PCB, delete a PCB, block a PCB, unblock a PCB, suspend a PCB, resume a PCB, change a PCB's priority, and list created PCBs. These PCB actions are selected through the flags.

The general form for executing a PCB command is as follows,

```
pcb [options] [process name] [suboptions] [suboption arguments]
```

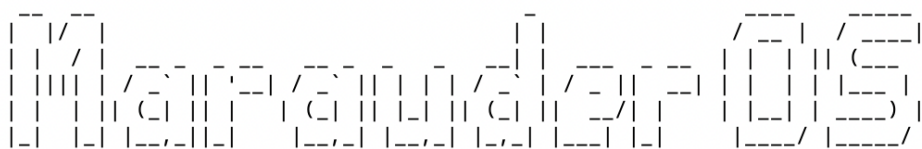
where the options indicate the user command to be run by the use of a flag, process name is the name of the PCB, suboptions are used to indicate the extra information needed in some commands, and the suboption arguments are the values of the extra information needed. The options are indicated with a hyphen then a single character, where the suboptions are indicated with double hyphens and the suboption.

### 3.1 Create a PCB

The `-c` flag under the `pcb` command within MarauderOS empowers users to initiate the creation of a PCB with specific attributes. With this command, users can define a custom name for the PCB, designate its class to indicate whether it is a user process or a system process, and set its priority level. Upon execution, the newly created PCB is seamlessly integrated into the system's process management framework, specifically within the "ready not suspended" queue. This feature allows for the systematic creation and allocation of PCBs, streamlining the overall process control and enhancing the flexibility and control offered by MarauderOS.

#### Usage

To create a PCB, use the `-c` as the flag and then specify a process name. This process name must be at least 8 characters, the user will face an error message otherwise. After the process name is specified, the suboptions are then stated. The create PCB command requires two suboptions: `--class` and `--pri`. The order these are stated does not matter. The class can only take on one of two values: `user` to make a user process and `system` to make a system process. The priority takes an integer value from 0 to 9 inclusive. The user will face an error message if the suboptions are not specified correctly or the value is incorrect. An example usage can be seen in Figure 7.



```
Welcome to MarauderOS | Use 'help' command to see list of commands
[> pcb -c user_proc1 --class user --pri 1
INFO: PCB Created
[> pcb -c sys_proc1 --pri 3 --class system
INFO: PCB Created
> █
```

Figure 7: Example Usage of the `pcb -c` Command




## 3.2 Delete a PCB

The `-d` flag under the `pcb` command in MarauderOS provides users with a straightforward method to remove a PCB from the system. When executed, this command first identifies the queue in which the target PCB resides. It then proceeds to remove the PCB from that specific queue, effectively deleting it from the system's process management. This capability is valuable for users who need to terminate processes efficiently, ensuring that resources are released and system performance is optimized.

### Usage

To delete a PCB, use the `-d` as the flag and specify the process name. If the process name does not exist in any of the queues, the user will face an error message. This command does not require any suboptions. An example usage of the delete command can be seen in Figure 8.



```
Welcome to MarauderOS | Use 'help' command to see list of commands
[> pcb -c user_proc1 --class user --pri 1
INFO: PCB Created
[> pcb -d user_proc1
INFO: PCB deleted
> █
```

Figure 8: Example Usage of the `pcb -d` Command

## 3.3 Block a PCB

The `-b` flag within the `pcb` command in MarauderOS offers users halt and manage processes within the system. When invoked, this command initiates a series of actions to suspend a specified PCB. First, it removes the PCB from its current queue. Next, it alters the PCB's running state to "blocked," indicating that it's currently in a suspended state. Depending on the PCB's dispatching state, the command then places it into either the general "blocked" queue or

the "suspended blocked" queue, allowing for precise control over process suspension.

## Usage

To block a PCB, use the `-b` flag and specify the process name. If the process name does not exist in any of the queues, the user will face an error message. This command does not require any suboptions. An example usage of the delete command can be seen in Figure 9.

### 3.4 Unblock a PCB

The `-d` flag within the `pcb` command in MarauderOS provides users with a tool to re-activate and manage previously suspended processes within the system. When executed, this command initiates a series of actions to unblock a specified PCB. First, it removes the PCB from its current queue, ensuring it is no longer held in a suspended state. Next, it changes the PCB's running state to "unblocked," indicating that it's ready for execution. Depending on the PCB's dispatching state, the command then places it into either the general "ready" queue or the "suspended ready" queue, effectively preparing it for execution.

## Usage

To unblock a PCB, use the `-u` flag and specify the process name. If the process name does not exist in any of the queues, the user will face an error message. This command does not require any suboptions. An example usage of the delete command can be seen in Figure 9.



```

Welcome to MarauderOS | Use 'help' command to see list of commands
[> pcb -c user_proc1 --class user --pri 1
INFO: PCB Created
[> pcb -b user_proc1
INFO: PCB blocked
[> pcb -u user_proc1
INFO: PCB unblocked
[> █
```

Figure 9: Example Usage of the `pcb -b` and `pcb -u` Command

### 3.5 Suspend a PCB

The `-s` flag within the `pcb` command in MarauderOS grants users the capability to temporarily halt processes within the system. When invoked, this command performs a series of actions to suspend a designated PCB. Firstly, it removes the PCB from its current queue, effectively taking it out of active circulation within the system. Subsequently, it alters the PCB's dispatching state to "suspended," indicating that it is now in a state of suspension. Depending on the PCB's running state at the time of suspension, the command places it into either the "suspended ready" queue or the "suspended blocked" queue, ensuring that suspended processes are categorized appropriately for future reactivation or further management.

#### Usage

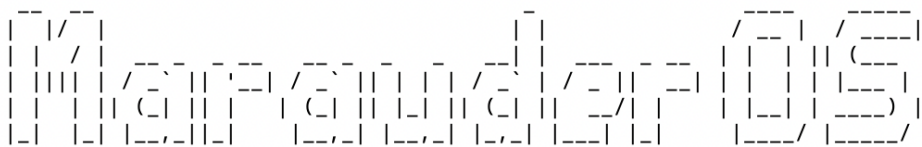
To suspend a PCB, use the `-s` flag and specify the process name. If the process name does not exist in any of the queues, the user will face an error message. A user cannot suspend a system process. This command does not require any suboptions. An example usage of the suspend command can be seen in Figure 10.

### 3.6 Resume a PCB

The `-r` flag within the `pcb` command in MarauderOS offers users a means to reactivate previously suspended processes within the system. When executed, this command initiates a series of actions to resume a specified Process Control Block (PCB). Firstly, it removes the PCB from its current queue, ensuring that it's no longer in a suspended state. Subsequently, it changes the PCB's dispatching state to "ready," signifying that it's prepared for execution. Depending on the PCB's running state at the time of resumption, the command places it into either the "suspended ready" queue or the "suspended blocked" queue, effectively categorizing it for further management or execution.

## Usage

To resume a PCB, use the `-r` flag and specify the process name. If the process name does not exist in any of the queues, the user will face an error message. This command does not require any suboptions. An example usage of the suspend command can be seen in Figure 10.



```
Welcome to MarauderOS | Use 'help' command to see list of commands
[> pcb -c user_proc1 --class user --pri 1
INFO: PCB Created
[> pcb -s user_proc1
INFO: PCB suspended
[> pcb -r user_proc1
INFO: PCB resumed
> █
```

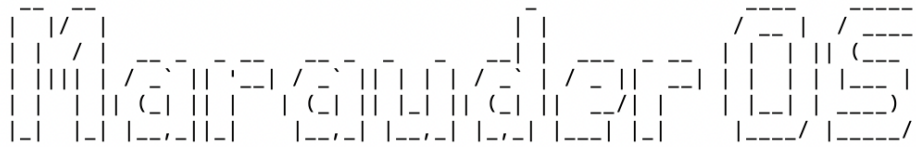
Figure 10: Example Usage of the `pcb -s` and `pcb -r` Command

## 3.7 Change a PCB's Priority

The `-p` flag within the `pcb` command in MarauderOS provides users with a tool to adjust the priority level of an existing process within the system. When invoked, this command facilitates the modification of a PCB priority setting, allowing users to fine-tune the importance or urgency of a specific process relative to others. By altering the priority, users can influence the process scheduling algorithm, affecting how CPU time is allocated among active processes.

## Usage

To change a PCB's priority, use the `-p` as the flag and then specify the process. If the process name does not exist in any of the queues, the user will face an error message. This command requires one suboption: `--pri`. The priority takes an integer values from 0 to 9 inclusive. The user will face an error message if the suboption or value is incorrect. An example usage can be seen in Figure 11.



```
Welcome to MarauderOS | Use 'help' command to see list of commands
[> pcb -c user_proc1 --class user --pri 1
INFO: PCB Created
[> pcb -p user_proc1 --pri 2
INFO: PCB priority changed
> █
```

Figure 11: Example Usage of the `pcb -p` Command

### 3.8 List Current PCBs

The `-l` command within MarauderOS offers users a versatile means to retrieve and display vital information about the existing PCBs within the system. This command provides users with the flexibility to customize the output based on their specific needs. Users can choose to list all PCBs, or selectively list PCBs based on their current state, such as "ready" or "blocked." Additionally, users can opt to list a single PCB, providing detailed information about a particular process.

#### Usage

To list information about PCBs, use the `-l` as the flag. If there is no argument specified, then all of the PCBs in the system will print to the terminal (as seen in Figure 12). If the user wants to only print just the ready processes, then use `ready` after the `-l` flag (as seen in Figure 13). If the user wants to see just the blocked processes, then use `blocked` after the `-l` flag (as seen in Figure 14). If the user wants to see information for just a single process, then specify the process name after the `-l` flag (as seen in Figure 15). If the user specifies a string after the flag that is not specified here, an error message is displayed. If the user specifies a process that does not exist, then they will see an error message.

```
[> pcb -l
```

| Name       | Class          | State   | Suspended | Status    | Priority |
|------------|----------------|---------|-----------|-----------|----------|
| user_proc5 | user process   | ready   | not       | suspended | 1        |
| sys_proc2  | system process | ready   | not       | suspended | 1        |
| sys_proc1  | system process | ready   | not       | suspended | 2        |
| sys_proc4  | system process | ready   | not       | suspended | 4        |
| user_proc3 | user process   | ready   | suspended |           | 2        |
| user_proc6 | user process   | ready   | suspended |           | 3        |
| user_proc2 | user process   | blocked | not       | suspended | 2        |
| user_proc4 | user process   | blocked | not       | suspended | 5        |
| sys_proc3  | system process | blocked | not       | suspended | 6        |
| user_proc1 | user process   | blocked | suspended |           | 1        |

Figure 12: Example Usage of the `pcb -l` Command — List All

| [> pcb -l ready |                |       |                  |          |
|-----------------|----------------|-------|------------------|----------|
| Name            | Class          | State | Suspended Status | Priority |
| user_proc5      | user process   | ready | not suspended    | 1        |
| sys_proc2       | system process | ready | not suspended    | 1        |
| sys_proc1       | system process | ready | not suspended    | 2        |
| sys_proc4       | system process | ready | not suspended    | 4        |
| user_proc3      | user process   | ready | suspended        | 2        |
| user_proc6      | user process   | ready | suspended        | 3        |

Figure 13: Example Usage of the `pcb -l` Command — List Ready

| [> pcb -l blocked |                |         |                  |          |
|-------------------|----------------|---------|------------------|----------|
| Name              | Class          | State   | Suspended Status | Priority |
| user_proc2        | user process   | blocked | not suspended    | 2        |
| user_proc4        | user process   | blocked | not suspended    | 5        |
| sys_proc3         | system process | blocked | not suspended    | 6        |
| user_proc1        | user process   | blocked | suspended        | 1        |

Figure 14: Example Usage of the `pcb -l` Command — List Block

```
MarauderOS
```

Welcome to MarauderOS | Use 'help' command to see list of commands

```
[> pcb -c user_proc1 --class user --pri 1  
INFO: PCB Created  
> pcb -l user_proc1
```

| Name       | Class        | State | Suspended Status | Priority |
|------------|--------------|-------|------------------|----------|
| user_proc1 | user process | ready | not suspended    | 1        |

```
>
```

Figure 15: Example Usage of the `pcb -l` Command — List PCB

### 3.9 Summary of PCB Commands

Seen below in Table 1 is a summary of the flags and their usage within the `pcb` command.

| Flag | Usage                 | Suboptions Required |
|------|-----------------------|---------------------|
| -c   | Create a PCB          | --class & --pri     |
| -d   | Delete a PCB          | N/A                 |
| -b   | Block a PCB           | N/A                 |
| -u   | Unblock a PCB         | N/A                 |
| -s   | Suspend a PCB         | N/A                 |
| -r   | Resume a PCB          | N/A                 |
| -p   | Change PCB's Priority | --pri               |
| -l   | List Current PCBs     | See Section 3.8     |

Table 1: Table of `pcb` Flags and Usage