Basil OS Version 6.0: User's Manual

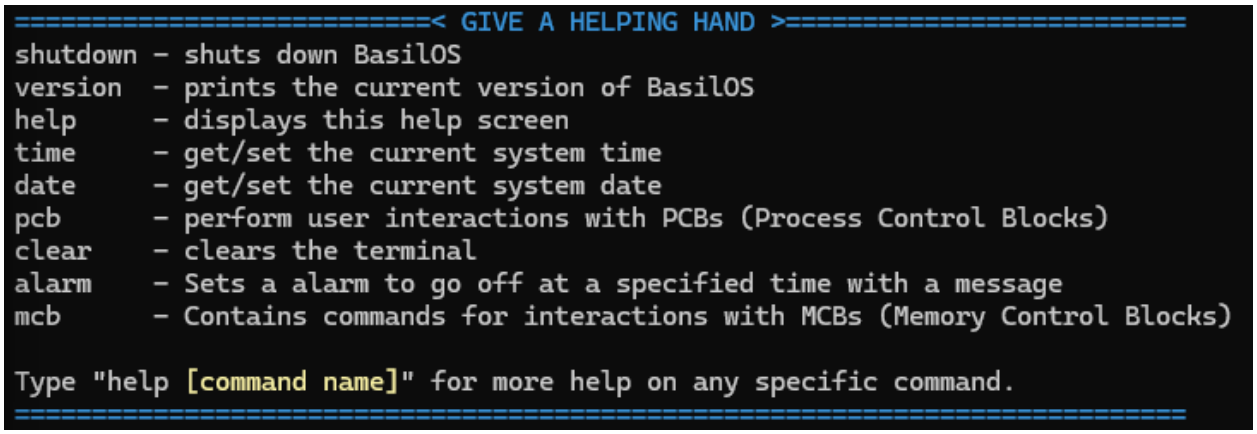
Table of Contents

Getting Started

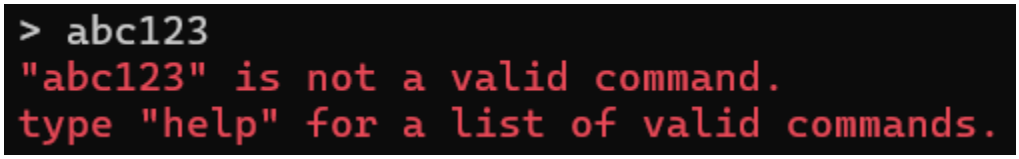
Welcome to the BasilOS User's Manual! This guide will show you what commands you have at your disposal and how to interact with BasilOS Version 5.0.

Upon the launch of BasilOS, you will be greeted by an ASCII Graphic and shown an initial message telling you about the "help" command.

To get started, type "help" into the terminal and hit enter on your keyboard to be shown a list of all commands. After completing that, your screen will look like this:



In the next section, we will go into more detail about each individual command, command variations, proper syntax, and possible errors that can occur. It should be noted that the following commands in this guide are the only valid inputs for the system. If you enter anything other than the commands listed or an incorrect command, you will be shown the message, "[Invalid input] is not a valid command. type "help" for a list of valid commands."

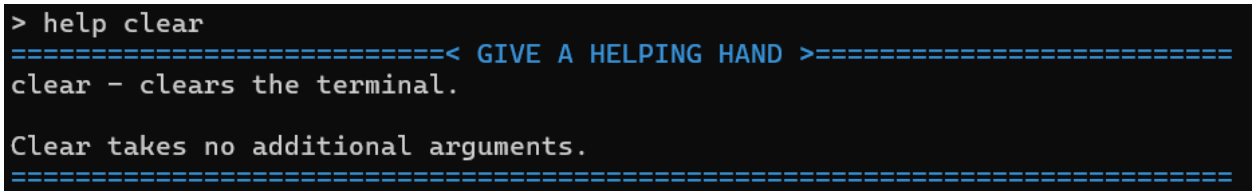


User Commands

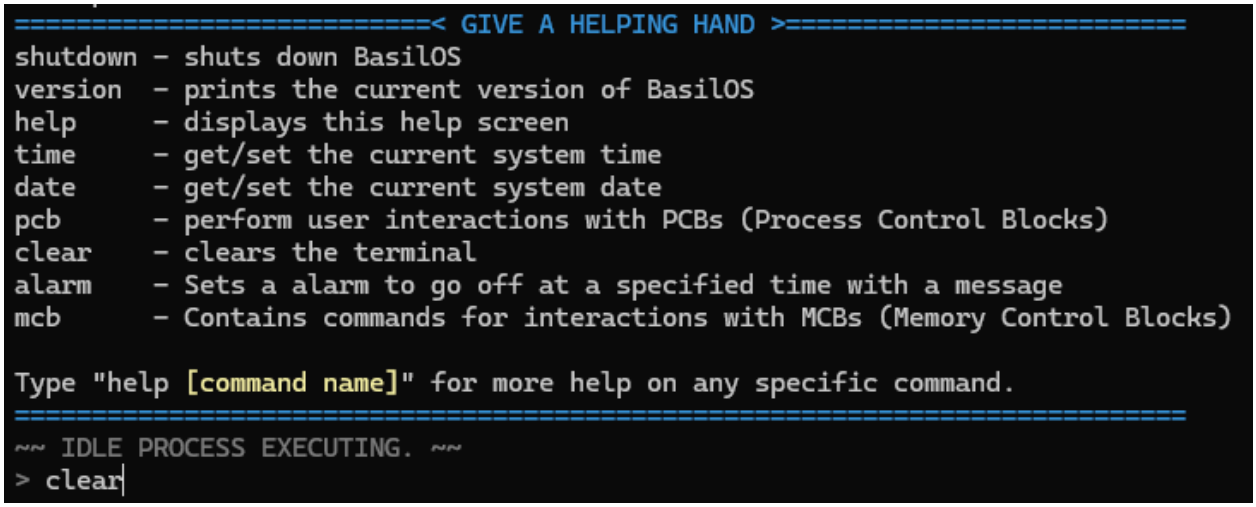
This section of the guide details all commands that are available to the user. Note that they are alphabetized for ease of searching.

1.) *clear* - The “clear” command is used to completely clear the terminal of text. The “clear” command takes no additional arguments.

A. Entering the “help clear” command will display additional command information



B. The second image shows typing “clear”, but not entering the command.



C. The third is after entering “clear”.

2.) *date* - The “date” command is used to get/set the current system date and has two optional additional arguments.

A. Entering “help date” will display additional command information

```
> help date
============================< GIVE A HELPING HAND >============================
date - gets/sets the system date.

Additional args:
date get            - print the system date to the screen
date set [MM/DD/YY] - set system date to MM/DD/YY
==============================================================================
```

B. Entering “date get” will print the current system date to the screen in the form of [MM]/[DD]/[YY].

```
> date get
02/27/25
```

C. Entering “date set [MM] [DD] [YY]” and exchanging the placeholders for a valid date will allow you to set the system date.

```
> date set 02/27/25
Date updated successfully.
```

D. Entering an invalid date will display the error message, “Invalid date values.”

```
> date set 13/45/96
Invalid date values.
```

E. Entering only “date set” will display the error message, “Incorrect formatting”.

```
> date set
Incorrect formatting.
Please type date as MM/DD/YY.
```

3.) *help* - The “help” command is used to display all commands available to the user.

A. Entering “help help” will display additional command information

```
> help help
============================< GIVE A HELPING HAND >============================
help - displays the help screen.

Additional args:
help [command name] - displays help for a specific command
==============================================================================
```

B. Entering the “help” command displays all the commands available to the user, along with a short description of what each command does.

```
> help
============================< GIVE A HELPING HAND >============================
shutdown - shuts down BasilOS
version  - prints the current version of BasilOS
help     - displays this help screen
time     - get/set the current system time
date     - get/set the current system date
pcb      - perform user interactions with PCBs (Process Control Blocks)
clear    - clears the terminal
alarm    - Sets a alarm to go off at a specified time with a message
mcb      - Contains commands for interactions with MCBs (Memory Control Blocks)

Type "help [command name]" for more help on any specific command.
==============================================================================
```

C. Ex. entering “help date” will display more information about the “date” command.

```
> help date
============================< GIVE A HELPING HAND >============================
date - gets/sets the system date.

Additional args:
date get            - print the system date to the screen
date set [MM/DD/YY] - set system date to MM/DD/YY
==============================================================================
```

4.) *pcb* - The “pcb” command is used to establish Process Control Blocks (PCBs) for tracking and managing processes.

A. Entering “help pcb” will display additional command information. There are 12 PCB command variations. The rest of section 4 goes over these commands in the order they appear below.

```
> help pcb
============================< GIVE A HELPING HAND >============================
pcb - user-interactions with the PCBs (Process Control Blocks).

Additional args:
pcb delete   [name*]                - delete a pcb
pcb suspend  [name*]                - suspends a pcb
pcb resume   [name*]                - unsuspends a pcb
pcb priority [name*] [priority**] - sets a pcbs priority
pcb show     [name*]                - Displays specific process
pcb loadR3   <suspended***>         - Creates 5 processes for testing contexts
pcb showready                       - Displays all ready processes
pcb showrunning                     - Displays all (1) running processes
pcb showblocked                     - Displays all blocked processes
pcb showall                         - Displays all processes
*name must be, at most, 8 characters long.
**priority must be between 0 and 9: 0 the highest priority, 9 the lowest.
***include "suspended" to load processes initially in suspended state.
==============================================================================
```

Deleting PCBs:

B. Entering “pcb delete [name]” will allow you to delete a PCB. If successful, “PCB deleted successfully.” will appear in the terminal.

```
> pcb delete proc1
PCB deleted successfully.
```

C. If you enter the command incorrectly, you will be shown “Incorrect formatting. Please type as: pcb delete [name].”

```
> pcb delete proc1 please
Incorrect formatting.
Please type as:pcb delete [name].
```

D. Entering the name of a PCB to delete that is typed incorrectly or doesn't exist will give you the error message, "PCB not found." Additionally, attempting to delete a PCB that has the class type "system" is not allowed. You will be given the error message "This pcb is a system PCB; users cannot delete it."

```
> pcb delete dnePCB
PCB not found.
```

```
> pcb delete sysPCB
This pcb is a system PCB; users cannot delete it.
```

Suspending PCBs:

E. Entering "pcb suspend [name]" will allow you to suspend a PCB.

```
> pcb suspend proc3
PCB suspended successfully.
```

F. If you enter the command incorrectly you will be shown the error message, "Incorrect formatting. Please type as: pcb suspend [name]".

```
> pcb suspend proc3 now
Incorrect formatting.
Please type as:pcb suspend [name].
```

G. Entering the name of a PCB to suspend that is typed incorrectly or does not exist will give you the error message, "PCB not found". Additionally, attempting to suspend a PCB that is of class "system" is not allowed, and you will be shown the error message, "This pcb is a system PCB; users cannot suspend it"

```
> pcb suspend dnePCB
PCB not found.
```

```
> pcb suspend sysPCB
This pcb is a system PCB; users cannot suspend it.
```

Resuming PCBs:

H. Entering "pcb resume [name]" will allow you to resume a PCB that has been suspended.

```
> pcb resume proc3
PCB resumed successfully.
```

I. If you enter the command incorrectly you will be shown the error message, “Incorrect formatting. Please type as: pcb resume [name].”

```
> pcb resume proc3 now
Incorrect formatting.
Please type as:pcb resume [name].
```

J. Entering the name of a PCB to resume that is typed incorrectly or does not exist will give you the error message, “PCB not found.”

```
> pcb resume dnePCB
PCB not found.
```

PCB Priority Levels:

K. Entering “pcb priority [name] [new priority]” will allow you to change the priority level of the PCB. The PCB priority level ranges from 0-9, with 0 being the highest priority, and 9 being the lowest.

```
> pcb priority proc4 3
PCB priority changed successfully.
```

L. Attempting to enter a PCB priority level outside the valid range will give you the error message, “PCB priority value is integer restricted to valued 0-9.”

```
> pcb priority proc4 12
PCB priority value is integer restricted to valued 0-9.
```

M. Entering the name of a PCB to change its priority that is typed incorrectly or does not exist will show you the error message, “PCB not found.”

```
> pcb priority dnePCB 3
PCB not found.
```

Displaying PCB Information:

N. Entering “pcb show [name]” will display information about a specific PCB including the name, class, state, suspended state, and priority level.

```
> pcb show proc5
        NAME            CLASS    STATE    SUSPENDED_STATE PRIORITY
        ================================================================
READY   :proc5          USER     READY    NOT SUSPENDED   5
```

O. Attempting to show a PCB that does not exist will display the error message, “PCB not found.”

```
> pcb show dnePCB
PCB not found.
```

Loading Processes:

P. Entering “pcb loadR3” will load the test processes from <processes.h>. They are loaded and queued in non-suspended ready state and were given descriptive names. Each process starts with a priority level of 5.

```
> pcb loadR3
5 out of 5 processes loaded
```

Q. Attempting to load the PCBs after they’re already loaded will display an error message to the screen:

```
> pcb loadR3
Couldn't create "proc1" because a PCB already exists with that name
Couldn't create "proc2" because a PCB already exists with that name
Couldn't create "proc3" because a PCB already exists with that name
Couldn't create "proc4" because a PCB already exists with that name
Couldn't create "proc5" because a PCB already exists with that name
No processes loaded due above to conflicts
```

Displaying Ready PCBs:

R. Entering “pcb showready” will display all the processes that are in the ready state, along with their name, class, suspended state, and priority level.

```
> pcb showready
        NAME            CLASS   STATE   SUSPENDED_STATE PRIORITY
        ================================================================
READY   :proc1          USER    READY   NOT SUSPENDED   5
        proc2           USER    READY   NOT SUSPENDED   5
        proc3           USER    READY   NOT SUSPENDED   5
        proc4           USER    READY   NOT SUSPENDED   5
        proc5           USER    READY   NOT SUSPENDED   5
```

Displaying Blocked PCBs:

S. Entering “pcb showblocked” will display all the processes that are in the blocked state, along with their name, class, suspended state, and priority level.

```
> pcb showblocked
        NAME              CLASS    STATE    SUSPENDED_STATE PRIORITY
        ================================================================
BLOCKED:proc5             USER     BLOCKED NOT SUSPENDED    5
```

T. If no processes are blocked, then none will be displayed.

```
> pcb showblocked
        NAME              CLASS    STATE    SUSPENDED_STATE PRIORITY
        ================================================================
BLOCKED:None!
```

Displaying Running PCBs:

U. Entering “pcb showrunning” will display all the processes that are in the running state, along with their name, class, suspended state, and priority level. Of course, there can only ever be one process running at a time, so only one process will ever show.

```
> pcb showrunning
        NAME              CLASS    STATE    SUSPENDED_STATE PRIORITY
        ================================================================
RUNNING:comhand           SYSTEM   RUNNING NOT SUSPENDED    0
```

Displaying all Processes:

V. Entering “pcb showall” (show all) will display all the PCBs, regardless of their state, along with their name, class, suspended state and priority level.

```
> pcb showall
        NAME              CLASS    STATE    SUSPENDED_STATE PRIORITY
        ================================================================
RUNNING:comhand           SYSTEM   RUNNING NOT SUSPENDED    0
READY  :proc1             USER     READY    NOT SUSPENDED    5
        proc4             USER     READY    NOT SUSPENDED    5
        proc5             USER     READY    SUSPENDED        5
        sysIDLE           SYSTEM   READY    NOT SUSPENDED    9
BLOCKED:proc2             USER     BLOCKED NOT SUSPENDED    5
        proc3             USER     BLOCKED SUSPENDED        5
```

5.) *shutdown* - The “shutdown” command shuts down the operating system.

A. Entering “help shutdown” will display additional command information

```
> help shutdown
============================< GIVE A HELPING HAND >============================
shutdown - shuts down BasilOS.
Will prompt the user to type y/n to confirm shutdown.

Shutdown takes no additional arguments.
=============================================================================
```

B. Entering “shutdown” into the system will begin the shutdown sequence and then prompt the user for confirmation.

```
> shutdown
Are you sure you want to shutdown? y/n
```

C. Entering ‘y’ will confirm the system shutdown and close the OS.

```
> shutdown
Are you sure you want to shutdown? y/n
> y
klogv: Starting system shutdown procedure...
klogv: Halting CPU...
```

D. Entering ‘n’ will abort the shutdown sequence. “Shutdown sequence terminated.” will be displayed and the operating system will resume operation.

```
> shutdown
Are you sure you want to shutdown? y/n
> n
Shutdown sequence terminated.
>
```

E. Entering invalid input during the shutdown sequence will reprompt the user until a valid input is entered.

```
> shutdown
Are you sure you want to shutdown? y/n
> a
Please enter y/n to confirm shutdown.
> b
Please enter y/n to confirm shutdown.
> y
klogv: Starting system shutdown procedure...
klogv: Halting CPU...
```

6.) *time* - The “time” command is used to get and set the system time.

A. Entering “help time” will display additional command information.

```
> help time
============================< GIVE A HELPING HAND >============================
time - gets/sets the system time.

Additional args:
time get            - print the system time to the screen
time set [HH:MM:SS] - set system time to HH:MM:SS, military time
=============================================================================
```

B. Entering the “time get” command will display the current time from the RTC to the user in the form of “HH:MM:SS”. It should be noted that the time is displayed in military format. (Ex. 17:34:01 = 5:34:01 pm)

```
> time get
17:34:01
```

C. Entering the “time set” command allows the user to change the time of the RTC by inputting their desired time in the form “time set [HH] [MM] [SS]”. (Ex. 13:27:15 = 1:27:15 pm)

```
> time set 13:27:15
Time updated successfully.
```

D. Entering an invalid time will display an error message, “Invalid time values.” Valid ranges are [0-23] for hours, [0-59] for minutes, [0-59] for seconds.

```
> time set 24:61:72
Invalid time values.
```

7.) *version* - The “version” command lists the OS version number in the major/minor style.

A. Entering “help version” will display additional command information

```
> help version
============================< GIVE A HELPING HAND >============================
version - prints the current version of BasilOS.

Version takes no additional arguments.
=============================================================================
```

B. Entering “version” will display the OS version number in major/minor style

```
> version
MPX Version 6.0
```

8.) *alarm* - The “alarm” command allows the user to create an alarm that will monitor the current time and display a message when that time has passed.

A. Entering the ‘help alarm’ command will display additional command information.

```
> help alarm
==========================< GIVE A HELPING HAND >==========================
alarm - creates a alarm process that will display a message at
a specified time

Additional args:
alarm [HH:MM:SS] [message] - sets alarm for time HH:MM:SS (military time)
that will display message.
=========================================================================
```

B. Entering ‘alarm HH:MM:SS Message’ will create the alarm at the supplied time that displays the supplied message.

```
> alarm 19:24:35 Awake!
Alarm created successfully.
```

```
ALARM (19:24:35): Awake!
```

C. Alarm messages can be multiword! Simply enclose a message in quotation marks.

```
> alarm 19:35:29 "Take the turkey out of the oven!"
Alarm created successfully.
```

```
ALARM (19:35:29): Take the turkey out of the oven!
```

D. Entering an alarm with an invalid time or any missing/malformated arguments will result in an error message being displayed, and no alarms being created. Valid time values range from [0-23] for hours, [0-59] for minutes, [0-59] for seconds. Every alarm must include a message.

```
> alarm 25:96:96 Wakedown!
Invalid time. Must be in 00:00:00..23:59:59.
```

```
> alarm 19 25 25 Wakeleft!
Time must be in HH:MM:SS format.
```

9.) *mcb* - The “mcb” command allows the user to allocate, free, and view memory control blocks.

A. Entering “help mcb” will display additional command information. There are 5 MCB command variations. The rest of section 8 will go over all of these commands in the order that they appear below.

```
> help mcb
==========================< GIVE A HELPING HAND >=========================
mcb - user-interactions with MCBs (Memory Control Blocks).

Additional args:
mcb showallocated        - Shows all allocated MCBs.
mcb showfree             - Shows all free MCBs.
mcb showall              - shows all MCBs.
========================================================================
```

B. Entering “mcb showallocated” will allow you to display all allocated memory blocks. It will print that in a table that looks like the following.

```
> mcb showallocated
        BLOCK NUMBER    START ADDRESS   BLOCK SIZE      ALLOCATED
        ================================================================
        0               0xd000014       1048            ALLOCATED
        1               0xd000440       1048            ALLOCATED
```

C. Entering “mcb showfree” will allow you to display all of the free memory blocks, it will print in a similar type of table shown below.

```
> mcb showfree
        BLOCK NUMBER    START ADDRESS   BLOCK SIZE      ALLOCATED
        ================================================================
        2               0xd00086c       47844           FREE
```

D. Entering “mcb showall” will allow you to display all MCBs. It looks something like this.

```
> mcb showall
        BLOCK NUMBER    START ADDRESS   BLOCK SIZE      ALLOCATED
        ================================================================
        0               0xd000014       1048            ALLOCATED
        1               0xd000440       1048            ALLOCATED
        2               0xd00086c       1048            ALLOCATED
        3               0xd000c98       1048            ALLOCATED
        4               0xd0010c4       1048            ALLOCATED
        5               0xd0014f0       1048            ALLOCATED
        6               0xd00191c       1048            ALLOCATED
        7               0xd001d48       42504           FREE
```