

# {nexTTeam} mpx module 4

User Manual

,

**{teaMMembers}**

*Teddy Hale*

*Jessica Hammersla*

*Mya Junkin*

*Philip Wenger*

## Table of Contents

<b>Overview</b>	<b>4</b>
<b>Beginning</b>	<b>4</b>
<b>Version</b>	<b>5</b>
<b>Date</b>	<b>5</b>
<b>Time</b>	<b>6</b>
<b>Shutdown</b>	<b>7</b>
<b>Help</b>	<b>7</b>
<b>Delete PCB</b>	<b>11</b>
<b>Suspend/Resume</b>	<b>11</b>
<b>Set Priority</b>	<b>12</b>
<b>Show PCBs</b>	<b>13</b>
<b>Yield</b>	<b>16</b>
<b>Loadr3</b>	<b>16</b>
<b>Alarms</b>	<b>16</b>
<b>Infinite</b>	<b>17</b>
<b>Module 5.1</b>	<b>18</b>
<b>Initialize Heap</b>	<b>18</b>
<b>Allocate Memory</b>	<b>19</b>
<b>Free Memory</b>	<b>19</b>
<b>Is Empty</b>	<b>20</b>
<b>Show Free</b>	<b>20</b>
<b>Show Allocated</b>	<b>21</b>
<b>Module 6</b>	<b>22</b>
<b>Help (Module 6 Version)</b>	<b>22</b>
<b>Quit</b>	<b>23</b>

<b>Print Boot Sector</b>	<b>23</b>
<b>Print Root Directory</b>	<b>24</b>
<b>Change Directory</b>	<b>25</b>
<b>List</b>	<b>26</b>
<b>Rename</b>	<b>27</b>
<b>Type</b>	<b>27</b>

## Overview

Welcome to the Next Team Operating System (NTOS)! *Module Four* introduces several essential features accessed through a menu-driven interface. This step in the NTOS development provides

- Alarm tools (with customizable messages)
- The Infinite Process
- Enhanced Process managing

The functionality provided with these features forms *NTOS Module Four*. Use this manual to guide you through your first use! The next section addresses how to begin.

## Beginning

This section outlines how to boot NTOS and what to do upon booting, whenever the user is not yet doing anything.

### To Boot:

In a Linux environment, the user opens the terminal and changes the directory to where NTOS is stored. The user enters

```
$ Make
```

Then enters

```
$ qemu-system-i386 -nographic -kernel kernel.bin -s
```

The user is now running NTOS

### The Menu:

Upon booting, the screen is blank. The user is able to enter commands at the top of the screen. If, at any point, the user is unsure of what options are available to them, they may either enter

```
Help
```

Which will display a list of menu options, or they may refer to this manual. Whenever the user is ready to do something, they may enter a command of their choice.

## Version

This section explains the use of the command “version.” To begin, the user enters

```
version
```

This command returns the following:

```
VERSION: 4.0
```

This indicates that *NTOS Module Four* is currently the version of NTOS in use.

## Date

This section explains the NTOS functionalities for setting and getting the system’s date.

### To Set the Date:

Setting the date changes the machine’s stored value for the current date. To begin setting the date, the user must enter

```
setDate
```

The user will then be prompted with the following message:

```
Enter date: mm/dd/yy
```

Beneath that, the user will enter the date in the format described by the prompt. For example, if the date to be entered is August 11th, 2019, the user will enter

```
08/11/19
```

If the user enters something in a format different than this, then nothing will happen and the user will be returned to the menu. If the correct format is used AND the date is valid, then the user will have set a new system date. The user will be returned to the menu.

### To Get the Date:

Getting the date lets the user see what value the machine has stored for the date. To begin getting the date, the user must enter

```
getDate
```

The system will then return the date in the following format:

```
mm/dd/yy
```

For example, if the stored values signified October 8th, 2006, the following would be returned:

```
11/08/06
```

The user will then be returned to the menu.

## Time

This section explains the NTOS functionalities for setting and getting the system's time.

### To Set the Time:

Setting the time changes the machine's stored value for the current time. To begin setting the time, the user must enter

```
setTime
```

The user will then be prompted with the following message:

```
Enter time: hh:mm:ss
```

Beneath that, the user will enter the time in the format described by the prompt (in military time). For example, if the time to be entered is 2:36 PM and 7 seconds, the user will enter

```
14:36:07
```

If the user enters something in a format different than this, then nothing will happen and the user will be returned to the menu. If the correct format is used AND the time is valid, then the user will have set a new system time. The user will be returned to the menu.

### To Get the Time:

Getting the time lets the user see what value the machine has stored for the time. To begin getting the time, the user must enter

```
getTime
```

The system will then return the time in the following format:

```
hh/mm/ss
```

For example, if the stored values signified 7:00 AM and 46 seconds, the following would be returned:

```
07:00:46
```

The user will then be returned to the menu.

## Shutdown

This section explains how to shutdown NTOS. To begin, the user will enter

```
shutdown
```

The user will then be prompted with

```
Are you sure you want to shut down? yes/no
```

If the user enters `yes`, the system will then shut down. If the user enters `no`, the user will be returned to the menu. If the user enters something else, the machine will prompt the user to

```
Please enter the word 'yes' or 'no'.
```

The user may then try entering `yes` or `no` again.

## Help

This section explains the use of the command “help.” To begin, the user enters

```
help
```

The user will then see

```
NAME
```

```
version - display current version of NTOS in use.
```

```
DETAIL DESCRIPTION
```

```
No further description
```

```
NAME
```

```
getTime - display current time of system.
```

```
DETAIL DESCRIPTION
```

```
Time will be displayed as hour:minute:second.
```

## NAME

setTime - change system's current time.

## DETAIL DESCRIPTION

Will prompt user to enter time as hh:mm:ss (i.e. hour:minute:second).

## NAME

getDate - display current date of system.

## DETAIL DESCRIPTION

Date will be displayed as month/day/year.

## NAME

setDate - change system's current date.

## DETAIL DESCRIPTION

Will prompt user to enter date as mm/dd/yy (i.e. month/day/year).

## NAME

setAlarm - set an alarm.

## DETAIL DESCRIPTION

Will prompt the user to enter a time as hh:mm:ss (i.e. hour:minute:second). Will also prompt user for a message to display when the alarm goes off.

## NAME

infinite - goes forever.

## DETAIL DESCRIPTION

Will create a process that sits idle forever and cannot be deleted unless it is suspended.

## NAME

Shutdown - shuts down NTOS.

## DETAIL DESCRIPTION

Will prompt user to confirm system shut down as yes/no.

## NAME

deletePCB - remove a PCB.

## DETAIL DESCRIPTION

Will find PCB, remove it from the queue, and free it.

## NAME

suspend - places PCB in suspended state.

## DETAIL DESCRIPTION



Will prompt user for a process name and will set to suspended state and move to the appropriate queue.

NAME

resume - places PCB in not suspended state.

DETAIL DESCRIPTION

Will prompt user for a process name and will set to not suspended state and move to the appropriate queue.

NAME

setPriority - sets PCB priority.

DETAIL DESCRIPTION

Will prompt user for a process name and ask for new priority and will set to new priority and move to the appropriate queue.

NAME

showPCB - display info for a PCB.

DETAIL DESCRIPTION

Will ask for a process name and display the process name, class, state, suspended status, and priority for a PCB.

NAME

showAllProcesses - display all PCBs.

DETAIL DESCRIPTION

Will display the process name, class, state, suspended status, and priority for all PCBs.

NAME

showReady - display all ready PCBs.

DETAIL DESCRIPTION

Will display the process name, class, state, suspended status, and priority for all ready queue PCBs.

NAME

showBlocked - display all blocked PCBs.

DETAIL DESCRIPTION

Will display the process name, class, state, suspended status, and priority for all blocked PCBs.

NAME

history - print command history.

DETAIL DESCRIPTION

Will display the user's ten previous commands.

## NAME

isEmpty - checks if the heap is empty.

## DETAIL DESCRIPTION

Will say whether any processes are taking up memory or not.

## NAME

yield - yield command handler.

## DETAIL DESCRIPTION

Will temporarily yield in R3.

## NAME

loadr3 - setup and load the five test processes.

## NAME

setAlarm - sets a user alarm.

## DETAIL DESCRIPTION

Will allow the user to enter hour, minute, and second for an alarm to set.

## NAME

initializeHeap - initializes memory heap.

## DETAIL DESCRIPTION

Will ask the user for memory to initialize heap in bytes.

## NAME

allocateMemory - allocate a block of memory.

## DETAIL DESCRIPTION

Will ask for block for how much memory to allocate in new block.

## NAME

freeMemory - frees a block of memory.

## DETAIL DESCRIPTION

Will ask the user for the start address of what block to free.

## NAME

isEmpty - says if memory is empty.

## DETAIL DESCRIPTION

Will say if the memory block is empty or not.

## NAME

showAllocated - show allocated block memories.

## DETAIL DESCRIPTION

Will show size and address of allocated memory blocks.

NAME

`showFree - show free block memories.`

DETAIL DESCRIPTION

`Will show size and address of free memory blocks.`

The user is then returned to the menu.

## Delete PCB

This section explains the NTOS functionalities for deleting an existing PCB for a process. To begin the process of deleting a PCB, the user must enter the command:  
`deletePCB`

The user will then be prompted with the following message:

`Enter the PCB name:`

Beneath that, the user will enter the name of the process they wish to delete.

If the user enters a name that does not match any process in the system, they will be given the following error.

There is no PCB by that name.

If the given name is valid, then the user will effectively remove this PCB from any queue and delete itself. The user will be returned to the menu.

## Suspend/Resume

This section explains the NTOS functionalities for suspending and resuming a process in the system.

### To Suspend a Process:

Suspending a process changes the PCB's stored value for its suspension state. To suspend a process, the user must enter

`suspend`

The user will then be prompted with the following message:

```
Enter the PCB name:
```

Beneath that, the user will enter the name of the process they wish to suspend.

If the user enters a name that does not match any process in the system, they will be given the following error.

```
There is no PCB by that name.
```

If the given name is valid, then the user will have set that process's suspension state to suspended. The user will be returned to the menu.

### To Resume:

Resuming a process changes the PCB's stored value for its suspension state. To resume a process, the user must enter

```
resume
```

The user will then be prompted with the following message:

```
Enter the PCB name:
```

Beneath that, the user will enter the name of the process they wish to resume.

If the user enters a name that does not match any process in the system, they will be given the following error.

```
There is no PCB by that name.
```

If the given name is valid, then the user will have set that process's suspension state to resume. The user will be returned to the menu.

## Set Priority

This section explains set a process's priority. To begin, the user will enter

```
setPriority
```

The user will then be prompted with

Enter the PCB name:

Beneath that, the user will enter the name of the process they wish to set priority for.

If the user enters a name that does not match any process in the system, they will be given the following error.

There is no PCB by that name.

If the given name is valid, then the user will be prompted again for the new priority number.

Enter the new priority for the PCB:

Beneath that, the user will enter the new priority for the PCB as an integer between 0 through 9 with 0 being of lowest priority. If a number outside of these bounds is entered the following error will be given:

The priority number is not valid

If the new number is valid a success message will, and the user will be returned to the menu.

## Show PCBs

### Show PCB:

To view one specific process's characteristics, the user must enter

showPCB

The user will then be prompted with the following message:

Enter the PCB name:

Beneath that, the user will enter the name of the process they wish to suspend.

If the user enters a name that does not match any process in the system, they will be given the following error.

There is no PCB by that name.

If the given name is valid, then the user will be shown that process's characteristics such as process name, class type (Application process or System process), state

(Ready, Running, or Blocked), suspension state (Suspended or Not Suspended), and priority number (an integer between 0-9 with 0 being lowest priority). This will be formatted as the example below:

```
Name: example
Class: System-Process
State: Ready
Suspend State: Not Suspended
Priority: 07
```

The user will then be returned to the menu.

### Show All Processes:

Showing all Processes shows the contents of all queues (Ready, Suspend-Ready, Blocked, and Suspend-Block) . To view all processes, the user must enter

```
showAllProcesses
```

The user will then view all processes within each queue in the format shown in the example below:

Ready Queue:

```
Name: dopey
Class: System-Process
State: Ready
Suspend State: Not Suspended
Priority: 07
```

Suspend Ready Queue:

```
Name: tootoo
Class: System-Process
State: Ready
Suspend State: Suspended
Priority: 02
```

Blocked Queue:

```
Name: example
Class: System-Process
State: Blocked
```

```
Suspend State: Not Suspended
Priority: 04
```

```
Suspend Blocked Queue:
```

The user will be returned to the menu.

### Show Ready Processes:

Showing Ready Processes displays the process characteristics of all processes in the Ready queue and Suspend-Ready queue. To show all ready processes, the user must enter

```
showReady
```

The user will then view all processes within each queue in the format shown in the example below:

```
Ready Queue:
```

```
Name: dopey
Class: System-Process
State: Ready
Suspend State: Not Suspended
Priority: 07
```

```
Suspend Ready Queue:
```

```
Name: tootoo
Class: System-Process
State: Ready
Suspend State: Suspended
Priority: 02
```

The user will be returned to the menu.

### Show Blocked Processes:

Showing Blocked Processes displays the process characteristics of all processes in the Blocked queue and Suspend-Blocked queue. To show all blocked processes, the user must enter

```
showBlocked
```

The user will then view all processes within each queue in the format shown in the example below:

Blocked Queue:

```
Name: example
Class: System-Process
State: Blocked
Suspend State: Not Suspended
Priority: 04
```

Suspend Blocked Queue:

The user will be returned to the menu.

## Yield

This section explains the use of the command “yield.” To begin, the user enters

```
version
```

This command then runs the next unsuspended ready process.

The user will be returned to the menu once all ready processes are finished.

## Loadr3

This section explains the use of the command “loadr3.” To begin, the user enters

```
loadr3
```

This command returns five consecutive success messages for suspending a PCB if successful.

This indicates that a PCB was created for each of the processes relating to module 3. They are loaded in as suspended-ready.

The user is immediately returned to the menu.

## Alarms

This section explains the NTOS functionalities for setting alarms and their resulting behavior.



Alarms in NTOS will “go off” after the time they are set for has passed, based on the internal clock. At that point, a message that is preset by the user will be displayed, alerting the user that the alarm has completed. There can be many alarms set at once.

### To Set an Alarm:

A user may set an alarm at any time and may set as many alarms as they may choose. To begin, the user enters:

```
setAlarm
```

The user will then be prompted with the following message:

```
Enter the time for the alarm: hh:mm:ss
```

Beneath that, the user will enter the time for the alarm in the format described by the prompt (in military time). For example, if the time to be entered is 2:36 PM and 7 seconds, the user will enter

```
14:36:07
```

After hitting “enter,” the user will next be prompted with:

```
Enter a message that's less than 100 characters which you want  
to display when the alarm goes off
```

The user may then enter a message of their choosing, although it must be under 100 characters. This message will be displayed to the user when the time set for the alarm has passed.

If the user enters a time in a format different than specified, then an error message will be sent and the user will be returned to the main menu. If the correct format is used AND the time is valid, then the user will have successfully set an Alarm. The user will be returned to the menu.

## Infinite

This section explains the use of the command “infinte.”

Infinite creates a process that does nothing forever. The only way to stop the infinite process is to, first, suspend it, and then remove it.

To begin, the user enters

```
infinite
```

This creates a process automatically without any additional user input, and the user is returned to the menu. Furthermore, this process will continue to be idle. Each time it runs, it will display this message:

```
Infinite Process Is Still Infinite
```

And then promptly return to its idle state.

## For Module 5.1

### Initialize Heap

This section explains the use of the command “initializeHeap.”

Initialize Heap creates a block of memory upon which further memory allocation functions can be used. It enables NTOS to simulate and use virtual memory.

To begin, the user enters

```
initializeHeap
```

The user will then be prompted with

```
Enter the size of the heap in bytes:
```

The user may choose any size up to 10,000 bytes, and enter it as so:

```
10000
```

The heap will then be created and the idle process will execute until another command is entered.

## Allocate Memory

This section explains the use of the command “allocateMemory.”

Allocate Memory accesses the heap created from initializing the heap and designates part of it as a “block” of memory. It requests a user-specified size for how big the block should be.

To begin, the user enters

```
allocateMemory
```

The user will then be prompted with

```
Enter the size of the block you want to allocate in bytes:
```

The user will then enter the desired number of bytes for this new block. If the user-entered size is too large, the following error will be displayed:

```
Allocation failed. No suitable memory block found.
```

Whether or not there is space for a new block of a user-specified size depends on how large the heap is, how much of that heap is already allocated, in what ways that heap is allocated, how large the user requests for the block to be, and other factors. If this error is displayed, consider these factors when attempting to allocate once again.

If the allocation is successful, the address offset will be displayed. For example, if the address offset is 24, this will be returned to the user:

```
Address offset: 24
```

Whether or not the allocation is successful, the user will be returned to the command prompt.

## Free Memory

This section explains the use of the command “freeMemory”

If a user desires to free a block of memory previously allocated, the user can use this command to do so if that block’s address is known. To begin, the user enters

```
freeMemory
```

The user will then be prompted with the following:

Enter the address for the block you want to free:

The user will then enter an address. If the address is invalid, or if there is no allocated memory at that memory address, the following message will be returned to the user:

Cannot free memory. No allocated memory to free.

If there is an allocated memory stored in the user-specified location, the user will be prompted with this message:

Memory Block has been freed.

The user will then be returned to the menu whether or not the user-specified address leads to a newly freed memory block.

## Is Empty

This section explains the use of the command “isEmpty”

Is Empty determines whether the heap created by the user is completely empty, or if it contains one or more allocated memory blocks.

To begin, the user enters

isEmpty

If the heap is, in fact, empty, the following message will be displayed:

The heap is completely free & empty

If the heap is not empty, as in there is one or more block of allocated memory, the following message will be displayed:

The heap is not empty

The user will then be returned to the menu.

## Show Free

This section explains the use of the command “showFree.” To begin, the user enters,

`showFree`

The user will then be presented with a table describing all the free blocks of memory contained in the heap. The heading of the table is

Free Memory:

Each free block’s location and size are shown. Each free block’s information is presented as follows, given that the block’s address is 58856 and size is 78177:

Address: 58856

Size: 78177

All the information from each free block is displayed in a list beneath the table. If there are no free blocks, there will be no information beneath the heading. After the table is displayed, the user will be returned to the menu.

## Show Allocated

This section explains the use of the command “showAllocated.” To begin, the user enters,

`showAllocated`

The user will then be presented with a table describing all the allocated blocks of memory contained in the heap. The heading of the table is

Allocated Memory:

Each allocated block’s location and size are shown. Each allocated block’s information is presented as follows, given that the block’s address is 58856 and size is 78177:

Address: 58856

Size: 78177

All the information from each allocated block is displayed in a list beneath the table. If there are no allocated blocks, there will be no information beneath the heading. After the table is displayed, the user will be returned to the menu.

## For Module 6

This module is separate from the rest of NTOS, and it is used to access and read disk image files. Module 6 is accessed by entering the folder “Module6” and entering the following commands. First, the user enters

```
gcc -o modulesix modulesix.c
```

Then, the user enters the program by entering

```
./modulesix cs450_2.img /file/path/to/folder/Module6
```

Where `cs450_2.img` is the disk image to be read from, and `/file/path/to/folder/Module6` is the absolute file path of the “Module6” folder on the computer.

The following menu will be displayed to the user:

```
Welcome to Next Team's File Management!  
Please type one of the available commands:  
quit  
help  
printBootSector  
printRootDirectory  
list  
rename  
type
```

The user may then enter any of the displayed commands.

## Help (Module 6 Version)

This section explains the Module 6 version of the “help” command. To begin, the user enters

```
help
```

The following will then be displayed to the user:

```
quit  
    Ends execution of module 6  
printBootSector  
    Prints out all information in the boot sector
```

`printRootDirectory`

Print all the info in the root directory

`changeDirectory <directoryName>`

Changes the directory to a subdirectory of the current directory

`list <file.extension>`

Lists the info for current directory by default, but lists for specified file if specified

`rename <oldFileName.extension> <newFileName.extension>`

Changes the file name

`type <fileName.extension>`

Display the file entered

The user will then be returned to the menu.

## Quit

This section explains the use of the Module 6 command “quit.” To begin, the user enters

`quit`

The user will be immediately returned to the terminal command prompt, and will need to restart Module 6 to access any of the other commands again.

## Print Boot Sector

This section explains the use of the Module 6 command “printBootSector.” It is used to display information regarding the boot sector of the disk in use. To begin, the user enters

`printBootSector`

The following information will then be displayed to the user in this format:

`printBootSector`

Bytes per Sector: 512

Sectors per Cluster: 1

Number of Reserved Sectors: 1

Number of FAT Copies: 2

Max Number of Root Directory Entries: 224

Total Number of Sectors in the File System: 2880

Number of Sectors per FAT: 9

Sectors per Track: 18

Number of Heads: 2

Total Sector Count for FAT32: 0

Boot Signature: 41

Volume ID: 236391907  
Volume Label: CS450\_2  
File System Type: FAT12

The user will then be returned to the menu.

## Print Root Directory

This section explains the use of the Module 6 command “printRootDirectory.”

The root directory of the disk is the starting point from where all files and all other directories can be accessed. This command displays information about the root directory regardless of the user’s current directory.

To begin, the user enters

```
printRootDirectory
```

The output will depend on the disk, but it will display information regarding each file or directory contained within the root directory in the following format:

```
File Name: CS450_2  
Extension:  
Attribute: 40  
Reserved: 0  
Creation Time: 00:00:00  
Creation Date: 8/19/1980  
Last Access Date: 8/19/1980  
Last Write Time: 22:22:22  
Last Write Date: 2/0/2016  
First Cluster: 0  
File Size: 0
```

```
File Name: 1984  
Extension: TXT  
Attribute: 32  
Reserved: 0  
Creation Time: 00:00:00  
Creation Date: 8/19/1980  
Last Access Date: 10/19/2016  
Last Write Time: 01:01:01  
Last Write Date: 2/0/2014  
First Cluster: 258
```



File Size: 598367

The first entry in the above list of two entries (separated with a blank line) is a directory, whereas the second is a TXT file.

After displaying this information, the user will be returned to the menu.

## Change Directory

This section explains the use of the Module 6 command “changeDirectory.”

This command allows the user to enter a directory different from the one the user is currently in. to begin, the user enters

`changeDirectory directoryName`

Where “directoryName” is the name of the directory to which the user would like to change. The user must specify a directory. If the specified directory doesn’t exist, the following error message is returned to the user:

`Did not find directory directoryName.`

If the specified directory is found, the following message is displayed:

`Successfully Changed Directory`  
`Printing current directory...`

The contents of the new directory are then displayed in the following format:

`File Name: CS450_2`  
`Extension:`  
`Attribute: 40`  
`Reserved: 0`  
`Creation Time: 00:00:00`  
`Creation Date: 8/19/1980`  
`Last Access Date: 8/19/1980`  
`Last Write Time: 22:22:22`  
`Last Write Date: 2/0/2016`  
`First Cluster: 0`  
`File Size: 0`

`File Name: 1984`  
`Extension: TXT`

Attribute: 32  
 Reserved: 0  
 Creation Time: 00:00:00  
 Creation Date: 8/19/1980  
 Last Access Date: 10/19/2016  
 Last Write Time: 01:01:01  
 Last Write Date: 2/0/2014  
 First Cluster: 258  
 File Size: 598367

The first entry in the above list of two entries (separated with a blank line) is a directory, whereas the second is a TXT file.

After displaying this information, or after displaying the error message, the user will be returned to the menu.

## List

This section explains the use of the Module 6 command “list,” which is used to display the contents of the current directory. To begin, the user enters

```
list filename.extension
```

Where “filename” is the name of the desired file, and “extension” is the extension of the desired file. The user may replace the filename or the extension with “\*” if the user doesn’t care to specify either one. In any of these cases, the given filename or extension (or lack thereof of either/both) is used to narrow down the contents that are listed to just those that match the requested filename and extension. If the user would like to see all contents of the current directory, to begin, the user may instead simply enter

```
list
```

Results will be displayed as follows:

File Name: CS450\_2  
 Extension:  
 Attribute: 40  
 Reserved: 0  
 Creation Time: 00:00:00  
 Creation Date: 8/19/1980  
 Last Access Date: 8/19/1980  
 Last Write Time: 22:22:22  
 Last Write Date: 2/0/2016

First Cluster: 0

File Size: 0

File Name: 1984

Extension: TXT

Attribute: 32

Reserved: 0

Creation Time: 00:00:00

Creation Date: 8/19/1980

Last Access Date: 10/19/2016

Last Write Time: 01:01:01

Last Write Date: 2/0/2014

First Cluster: 258

File Size: 598367

The first entry in the above list of two entries (separated with a blank line) is a directory, whereas the second is a TXT file.

The user will then be returned to the menu.

## Rename

This section explains the use of the Module 6 command “rename.” This command is used to change the name of a file. To begin, the user enters

```
Rename oldName newName
```

Where “oldName” is the name of the file to begin with, and “newName” is what the user wants the file’s name to change to. If the renaming is successful, then the following message will be displayed:

```
File Name Successfully Changed
```

If no file has the name “oldName,” then the following message will be displayed:

```
File name cannot be changed
```

The user will then be returned to the menu.

## Type

This section explains the Module 6 command “type.” Type allows the user to display a file. To begin, the user enters

type fileName.extension

Where “fileName” is the file’s name, and “extension” is the extension of the file. If the user-specified file doesn’t exist, the following message is displayed before returning the user to the menu:

The file entered was not found in the current directory

Here’s an example of what is displayed when this command is called, when the file entered is 1984.TXT :

type 1984.TXT

Title: Nineteen eighty-four

Author: George Orwell (pseudonym of Eric Blair) (1903-1950)

\* A Project Gutenberg of Australia eBook \*

eBook No.: 0100021.txt

Language: English

Date first posted: August 2001

Date most recently updated: November 2008

Project Gutenberg of Australia eBooks are created from printed editions which are in the public domain in Australia, unless a copyright notice is included. We do NOT keep any eBooks in compliance with a particular paper edition.

Copyright laws are changing all over the world. Be sure to check the copyright laws for your country before downloading or redistributing this file.

This eBook is made available at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg of Australia License which may be viewed online at <http://gutenberg.net.au/licence.html>

To contact Project Gutenberg of Australia go to <http://gutenberg.net.au>

The user is then returned to the menu.