MPX-OS User Manual

B. Burwell and A. Morash

Fall 2013

Contents

| 1 | Ove | erview | of COMHAN | 1 |
|---|-----|---------|-----------------------------|---|
| | 1.1 | The C | Command Handler | 1 |
| | 1.2 | A Not | te on Errors | 1 |
| | | | | |
| 2 | Sun | nmary | of Commands | 3 |
| 3 | Det | ailed I | Description of Each Command | 5 |
| | 3.1 | | - lon | 5 |
| | | 3.1.1 | Syntax | 5 |
| | | 3.1.2 | Usage | 5 |
| | | 3.1.3 | Example | 5 |
| | | 3.1.4 | Possible Errors | 5 |
| | 3.2 | date | | 6 |
| | | 3.2.1 | Syntax | 6 |
| | | 3.2.2 | Usage | 6 |
| | | 3.2.3 | Example | 6 |
| | | 3.2.4 | Possible Errors | 6 |
| | 3.3 | direc | ctory | 6 |
| | | 3.3.1 | Syntax | 7 |
| | | 3.3.2 | Usage | 7 |
| | | 3.3.3 | Example | 7 |
| | | 3.3.4 | Possible Errors | 7 |
| | 3.4 | stop | | 7 |
| | | 3.4.1 | Syntax | 7 |
| | | 3.4.2 | Usage | 7 |
| | | 3.4.3 | Example | 8 |
| | | 3.4.4 | Possible Errors | 8 |
| | 3.5 | help | | 8 |
| | | - | Syntax | Q |

ii CONTENTS

| | 3.5.2 | Usage | | | | | | | | | | 8 |
|------|--------|-----------------|--|--|--|--|--|--|--|--|--|----|
| | 3.5.3 | Example | | | | | | | | | | 8 |
| | 3.5.4 | Possible Errors | | | | | | | | | | 9 |
| 3.6 | prompt | 5 | | | | | | | | | | 9 |
| | 3.6.1 | Syntax | | | | | | | | | | 10 |
| | 3.6.2 | Usage | | | | | | | | | | 10 |
| | 3.6.3 | Example | | | | | | | | | | 10 |
| | 3.6.4 | Possible Errors | | | | | | | | | | 10 |
| 3.7 | alias | | | | | | | | | | | 10 |
| | 3.7.1 | Syntax | | | | | | | | | | 10 |
| | 3.7.2 | Usage | | | | | | | | | | 10 |
| | 3.7.3 | Example | | | | | | | | | | 10 |
| | 3.7.4 | Possible Errors | | | | | | | | | | 11 |
| 3.8 | show . | | | | | | | | | | | 11 |
| | 3.8.1 | Syntax | | | | | | | | | | 11 |
| | 3.8.2 | Usage | | | | | | | | | | 11 |
| | 3.8.3 | Example | | | | | | | | | | 11 |
| | 3.8.4 | Possible Errors | | | | | | | | | | 11 |
| 3.9 | load . | | | | | | | | | | | 12 |
| | 3.9.1 | Syntax | | | | | | | | | | 12 |
| | 3.9.2 | Usage | | | | | | | | | | 12 |
| | 3.9.3 | Example | | | | | | | | | | 12 |
| | 3.9.4 | Possible Errors | | | | | | | | | | 12 |
| 3.10 | resume | e | | | | | | | | | | 13 |
| | 3.10.1 | Syntax | | | | | | | | | | 13 |
| | 3.10.2 | Usage | | | | | | | | | | 13 |
| | 3.10.3 | Example | | | | | | | | | | 13 |
| | 3.10.4 | Possible Errors | | | | | | | | | | 13 |
| 3.11 | run | | | | | | | | | | | 14 |
| | 3.11.1 | Syntax | | | | | | | | | | 14 |
| | 3.11.2 | Usage | | | | | | | | | | 14 |
| | 3.11.3 | Example | | | | | | | | | | 14 |
| | 3.11.4 | Possible Errors | | | | | | | | | | 14 |
| 3.12 | susper | nd | | | | | | | | | | 14 |
| | | Syntax | | | | | | | | | | 15 |
| | 3.12.2 | Usage | | | | | | | | | | 15 |
| | 3.12.3 | Example | | | | | | | | | | 15 |
| | 3.12.4 | Possible Errors | | | | | | | | | | 15 |
| 3.13 | termin | nate | | | | | | | | | | 15 |
| | 3.13.1 | Syntax | | | | | | | | | | 15 |

| CONTENTS | | iii |
|-------------------|-------------------|-----|
| 3.13.2 | Usage | 15 |
| 3.13.3 | Example | 16 |
| 3.13.4 | Possible Errors | 16 |
| 3.14 setpr | iority | 16 |
| 3.14.1 | Syntax | 16 |
| 3.14.2 | Usage | 16 |
| 3.14.3 | Example | 16 |
| 3.14.4 | Possible Errors | 16 |
| $3.15~{ m clock}$ | | 17 |
| 3.15.1 | Syntax | 17 |
| 3.15.2 | Usage | 17 |
| 3.15.3 | Example | 17 |
| 3.15.4 | Possible Errors | 17 |
| 4 Summary | of Error Messages | 19 |

iv CONTENTS

Chapter 1

Overview of COMHAN

1.1 The Command Handler

All of your interactions with MPX-OS will be through the "Command Handler," referred to as "comhan." The command handler is a process that is loaded when MPX-OS first starts and allows users to execute commands such as getting help, running programs, setting the date, and other systemwide functionality.

To use the command handler, you must only start MPX. You will see the prompt mpx> which indicates that the system is ready to accept commands. You may then type in a command on the keyboard and press return. The command handler will then attempt to parse your command. If what you have entered is a valid system command, MPX will execute your request and then prompt you for the next command. If the text you enter at the MPX prompt is not valid, you will see an error message.

Note that the command handler parses your input case insensitively; that is, typing version, VERSION, and vERsiOn at the prompt will result in identical output.

All of the commands used in MPX-OS are listed in 2. Alternately, when in MPX-OS, the command help will display a list of commands and a brief usage description. Of particular note, however, is the stop command, which will terminate MPX-OS and return control of the computer to DOS.

1.2 A Note on Errors

When writing MPX-OS, we have subscribed to the philosophy that the user does not need to know what the system is doing unless it pertains to their actions. Thus, for the most part, the normal behavior of a command is to see no output. For example, when changing the prompt (see p. 9), there is no "prompt changed" message; the user should be able to trust the system to perform correctly.

Thus, messages will generally only be displayed when some unexpected behavior has occurred. For full and more precise details on exactly what the output of each command, see the appropriate documentation in this Manual.

Chapter 2

Summary of Commands

```
version
               Page 5
                        Prints the currently running version of MPX-OS.
       date
               Page 6
                        Prints or updates the system date.
               Page 6
                        Prints a listing of MPX programs that may be run.
  directory
               Page 7
       stop
                        Stops MPX-OS and returns to DOS.
               Page 8
                        Prints help on MPX commands.
       help
               Page 9
                        Changes the MPX prompt.
     prompt
                        Adds an alias for an MPX command.
      alias
              Page 10
              Page 11
                        Displays a table of MPX processes.
       show
              Page 12
       load
                        Loads a program.
              Page 13
                        Resumes a suspended program.
     resume
              Page 14
                        Runs a program.
        run
              Page 14
                        Puts a program into the suspended state.
    suspend
  terminate
              Page 15
                        Terminates a program, freeing its resources.
setpriority
              Page 16
                        Sets the execution priority for a program.
      clock
             Page 17
                        Prints or updates the system time.
```

Chapter 3

Detailed Description of Each Command

3.1 version

This command will display some information about the current version of MPX that is running.

3.1.1 Syntax

The command has the form version.

3.1.2 Usage

The command can be called from the MPX prompt at any time.

3.1.3 Example

1. To get the version of MPX-OS that is currently running:

Input: version

Output: MPX OS - Version 1.0

3.1.4 Possible Errors

There are no error conditions.

3.2 date

This command is used to retrieve and set the system date. Note that the date does not actually increment as time passes.

3.2.1 Syntax

The command optionally takes a date string formatted as mm/dd/yyyy where mm is a two-digit one-based integer representing the month (01 corresponding to January, 02 to February, and so on), dd similarly corresponding to a one-based date of month, and yyyy a year.

The command has the form date [mm/dd/yyyy].¹

3.2.2 Usage

The command can be called from the MPX prompt at any time.

3.2.3 Example

1. To get the current system date:

Input: date

Output: 12/04/2013

2. To set the system date: Input: date 12/04/2013

Output: None.

3. Attempting to set an invalid date:

Input: date 99/99/9999 Output: Invalid date.

3.2.4 Possible Errors

The message "Invalid date" will be displayed if the date is not in a valid format, i.e. the month is not between 1 and 12 or date not between 1 and 31, year not a number.

3.3 directory

The directory command will display a listing of all available MPX programs that can be loaded and run, along with their sizes.

¹Note that in this document, we use [] to denote optional parameters.

3.4. STOP 7

3.3.1 Syntax

The command has the form directory.

3.3.2 Usage

The command can be called from the MPX prompt at any time.

3.3.3 Example

1. Get a listing of programs when there are none:

Input: directory

Output: tt You have no programs

2. Get a listing of programs when there is one:

Input: directory

Output: tt You have 1 program

3. Get a listing of programs when there are four:

Input: directory

Output: tt You have 4 programs

3.3.4 Possible Errors

This command will not generate any errors.

3.4 stop

The stop command allows the user to exit MPX-OS and return to DOS. When the command is issued, the user will be prompted to confirm their action by typing the character y. If any other character is entered, the user will remain in MPX-OS.

3.4.1 Syntax

The command has the form stop.

3.4.2 Usage

The command can be called at any time through the MPX prompt.

3.4.3 Example

1. User calls stop, but does not want to exit:

Input: stop

Output: Are you sure you want to exit? [y/n]:

Input: n

Output: None.

2. User calls stop, and does confirm wanting to exit:

Input: stop

Output: Are you sure you want to exit? [y/n]:

Input: y

Output: ** COMHAN execution complete **

3.4.4 Possible Errors

There are no error conditions.

3.5 help

The help command allows the user to get help about using MPX commands. If the command is run with no options, a brief description of all commands. Alternately, if called with the name of a command as an option, it will give only the description of that command.

3.5.1 Syntax

The command has the form help [command].

3.5.2 Usage

The command can be called at any time through the MPX prompt.

3.5.3 Example

1. User wants help for all processes:

Input: help Output:

3.6. PROMPT 9

Name Use version Display version number help Provide information about commands directory List .mpx files date [mm/dd/yyyy] Display or set the system date Terminate execution of COMHAN stop prompt string Change the prompt for commands alias command=string Create an alias for a command Prints PCB information show allocate Builds PCB with specified options Frees the PCB called name free name load name[=ppp] Creates a process called name with priority ppp Resumes the process called name resume name Runs a process called name with priority ppp run name[=ppp] suspend name Suspends the process called name Terminates the process called name terminate name setpriority name=ppp Sets the priority of process name dispatch Runs each process once Perform clock operations clock [stop|start]

2. User wants help for the show command:

Input: help show

Output:

Name Use ------

show Prints PCB information

3.5.4 Possible Errors

There are no error conditions for this command.

3.6 prompt

This command is used to alter the prompt that is displayed to the user by the command handler. For example, the user could change the prompt from the default mpx> to >>.

3.6.1 Syntax

The command has the form prompt string, where string is the new prompt.

3.6.2 Usage

The command can be called from the MPX prompt at any time.

3.6.3 Example

1. User wants to set the prompt to "os":

Input: prompt os
Output: No output.

3.6.4 Possible Errors

If a string is not passed, the user will receive the error "No prompt specified."

3.7 alias

The alias command allows the user to make an alias for any of the commands. For example, if the user needs to access the value of the system clock in rapid succession, the clock command could be aliased to simply c.

3.7.1 Syntax

The syntax for this command is: alias command=new, with command the name of the command to be aliased, and new the string that should be made equivalent to command.

3.7.2 Usage

The command can be called from the MPX prompt at any time.

3.7.3 Example

1. Setting "c" to be an alias for "clock":

Input: alias clock=c

Output: None.

3.8. SHOW 11

3.7.4 Possible Errors

If command or new are not specified, the user will receive the error "No prompt specified."

3.8 show

The show command is used to view the status of various processes.

3.8.1 Syntax

The command is of the form show type, where type is one of free, all, system, application, suspended, ready, or init.

3.8.2 Usage

free Display all free process control blocks.

all Display all process control blocks.

system Display all system processes.

application Display all application processes.

suspended Display all suspended processes.

ready Display all ready processes.

init Display all processes that are waiting for I/O.

If a type is not given by the user, a message describing the possible values is displayed.

3.8.3 Example

1. User wants a listing of all process control blocks:

Input: show all

Output: A table of data about all process control blocks.

3.8.4 Possible Errors

If the type specified is not one of the valid types, an empty table will be displayed.

3.9 load

This command loads a program file into memory and prepares it for execution by MPX-OS.

3.9.1 Syntax

The command has the form load proc[=ppp] where proc is the name of the process to be loaded (and stored in the file proc.mpx), and ppp is an optional execution priority (default is zero). The priority p must satisfy -127 .

3.9.2 Usage

The command can be called from the MPX prompt at any time.

3.9.3 Example

1. User wants to load the process "proc1":

Input: load proc1
Output: None.

2. User wants to load the process "proc2" with priority 100:

Input: load proc2=100

Output: None.

3.9.4 Possible Errors

Several errors and warnings can occur when calling the command. If there is no file named proc.mpx where proc is the name argument to the load command, the message "Error: No program found with that name" will be displayed and the program will not be loaded.

If the program file does exist, but MPX-OS is unable to allocate sufficient memory for it to be loaded, the message "Error: Insufficient memory" will be displayed and the program will not be loaded.

If there is sufficient memory, but the system encounters an error while loading the program file into memory, the message "Error: Could not load program into memory" will be displayed, and again, the program will not be loaded.

If an invalid priority was specified by the user, the message "Warning: invalid priority specified, using default" will be displayed. MPX-OS will continue to load the program, but with zero as the priority.

3.10. RESUME 13

If there are no available process control blocks to allocate to the process, MPX cannot run the program at this time. In this case, the message "Error: No free PCBs" will be displayed and the program will not be loaded.

In the case that there is an error building the process control block for the process that has been loaded into memory, MPX-OS will not be able to execute it. Thus, the memory will be deallocated and the message "Error: Unable to build PCB" will be displayed to the user.

3.10 resume

The resume command allows a suspended process to continue running.

3.10.1 Syntax

The resume command has the format resume name | *.2 If a name is given, MPX-OS will attempt to find a process named name and resume it. Otherwise, if * is given, MPX-OS will resume all currently suspended processes.

3.10.2 Usage

The resume command is used when the user has suspended a process and wishes it to resume its execution.

3.10.3 Example

1. User wants to resume the process proc:

Input: resume proc Output: None.

2. User wants to resume all processes:

Input: resume *
Output: None.

3.10.4 Possible Errors

When **resume** is called with no options, a message reminding the user to enter the name of a process to resume or an asterisk is displayed.

If a name was given as the option, but MPX-OS cannot find a process with that name, the message "No process named x" will be displayed.

Upon correct execution, there will be no output.

²Note that we are using | to denote "or."

3.11 run

The run command behaves similarly to the load command, with the difference being quite minor. Whereas the load command simply allocated memory and set up a process control block for a process to be later run by a user using the resume command, the run command compresses these steps into one operation. Put simply, the run command does just that: it load a program from the disk and runs it.

3.11.1 Syntax

The command has the form run proc[=ppp] where proc is the name of the program (i.e., there is a file named proc.mpx) and ppp is the priority that should be used, where ppp satisfies the condition -127 < ppp < 127.

3.11.2 Usage

Usage

3.11.3 Example

1. User wants to run the process "proc1":

Input: run proc1 Output: None.

2. User wants to run the process "proc2" with priority 100:

Input: run proc2=100

Output: None.

3.11.4 Possible Errors

Since the run command first loads the process, all of the error messages from the load command can be obtained by using the run command.

In addition, if there was an error loading the process, the message "Error: process did not load correctly" will be displayed.

3.12 suspend

The suspend command is used when a running program needs to be temporarily paused.

3.13. TERMINATE 15

3.12.1 Syntax

The command has the form suspend proc|* where proc is the name of the program to be suspended. If the * option is used, all running application processes will be suspended.

3.12.2 Usage

The command can be called whenever there is a running program.

3.12.3 Example

1. User wants to suspend the process proc:

Input: suspend proc

Output: None.

2. User wants to suspend all processes:

Input: suspend * Output: None.

3.12.4 Possible Errors

If there is no process named proc, the message "No process with the specified name" will be displayed.

3.13 terminate

The terminate command frees the PCB that was associated with an application process and frees the memory that was in use by the program.

3.13.1 Syntax

Similarly to the resume and suspend commands, the format is terminate name |*, where name corresponds to the name of the application process to be terminated, and * to the fact that all application processes should be terminated.

3.13.2 Usage

The command can be run when there are currently running or suspended application processes.

3.13.3 Example

1. User wants to terminate the process proc:

Input: terminate proc

Output: None.

2. User wants to terminate all processes:

Input: terminate * Output: None.

3.13.4 Possible Errors

If there is no process called name, the message "No process with the specified name" will be displayed.

3.14 setpriority

The setpriority command allows the user to specify an execution priority for a process.

3.14.1 Syntax

The format of the command is setpriority name=ppp where name refers to the name of the affected process and ppp to the priority. As only the priority of application processes can be set, -127 < ppp < 127 must be satisfied.

3.14.2 Usage

The command can be called when there is a non-zero number of application processes in MPX.

3.14.3 Example

1. Set the priority of programm proc to be -90:

Input: setpriority proc=-90

Output: None.

3.14.4 Possible Errors

There are two possible errors from this command. First, the priority could be invalid, in which case the message "Error: invalid priority" will be displayed.

3.15. CLOCK 17

In the other, there is not really any process with the name specified, and MPX-OS will display "Error: invalid process name."

3.15 clock

The clock command is used to perform various tasks using the system clock. The clock can be stopped and started, set to a particular value, or have its value read.

3.15.1 Syntax

The command uses the form clock job|time, where job is one of stop, start and time is a string formatted as hh:mm:ss with hh hours, mm minutes, and ss seconds.

3.15.2 Usage

The clock command can be called at any time from the MPX-OS command handler.

3.15.3 Example

1. Starting the clock:

Input: clock start Output: None.

 Stopping the clock: Input: clock stop Output: None.

3. Getting the clock value:

Input: clock

Output: The current time is 21:09:44

4. Setting the system clock: Input: clock 21:09:44

Output: None.

3.15.4 Possible Errors

If for some reason, the clock is unable to be set, the message "Error setting clock" will be displayed.

Chapter 4

Summary of Error Messages

| Error | Command | Section | Page |
|--|-------------|---------|------|
| Invalid date | date | 3.2 | 6 |
| No prompt specified | prompt | 3.6 | 9 |
| No prompt specified | alias | 3.7 | 10 |
| Error: No program found with that name | load | 3.9 | 12 |
| Error: Insufficient memory | load | 3.9 | 12 |
| Error: Could not load program into memory | load | 3.9 | 12 |
| Warning: invalid priority specified, using default | load | 3.9 | 12 |
| Error: No free PCBs | load | 3.9 | 12 |
| Error: Unable to build PCB | load | 3.9 | 12 |
| No process named x | resume | 3.10 | 13 |
| Error: process did not load correctly | run | 3.11 | 14 |
| No process with the specified name | suspend | 3.12 | 14 |
| No process with the specified name | terminate | 3.13 | 15 |
| Error: invalid priority | setpriority | 3.14 | 16 |
| Error: invalid process name. | setpriority | 3.14 | 16 |
| Error setting clock | clock | 3.15 | 17 |

Index

```
{\tt alias},\,10
{\tt clock},\,17
Command Handler, 1
date, 6
{\tt directory},\, 6
\mathtt{help},\, 8
\verb"load", 12
MPX-OS, 1
\mathtt{prompt},\,9
\mathtt{resume},\,13
run, 14
\mathtt{setpriority},\,16
show, 11
stop, 7
\mathtt{suspend},\,14
{\tt terminate},\,15
{\tt version},\, 5
```