

Paul Prince's Multi-Programming Executive

User's Manual

(For Version R1)

Table of Contents

Introduction.....	3
Using the MPX Shell.....	4
Overview.....	4
MPX Shell Commands.....	4
The “help” Command.....	4
The “commands” Command.....	5
The “exit” Command.....	5
The “version” Command.....	6
The “date” Command.....	6
The “ls” Command.....	7

Introduction

The Multi-Programming Executive (MPX) is a project undertaken by students of Computer Science 450 at West Virginia University. It is similar to an operating system, but does not implement nearly all of the features and functions required of a complete (even if primitive) operating system; instead it is expected to run on top of Microsoft DOS or Windows.

The project is iterative, so this version of this MPX is not yet feature-complete.

This manual is intended to allow a hypothetical user with no previous knowledge of the MPX project the ability to use MPX system.

Like the rest of the project, this manual will continue to improve as the course progresses.

Getting Started

Obtaining MPX

The source code for this MPX is available online via Git; to download it, see:

- <http://github.com/pprince/cs450/tree/master/mpx>

Currently, binary (ready-for-use) versions are not distributed.

System Requirements

This MPX is designed to run on Microsoft Windows XP. It may work without modification on MS-DOS or on more recent versions of Microsoft Windows, but such configurations are untested and unsupported.

Starting MPX

To start MPX, simply run the `mpx.exe` program. This program accepts no arguments, but any arguments given will be ignored (i.e., no error will be printed about them.)

If all goes well, you will find yourself at the MPX shell prompt; see the next section for where to go from here!

Caveat: Current Working Directory

You must invoke MPX with the correct working directory! MPX expects that it will be run from the same directory that contains the `mpx.exe` file; it references other files using relative path names.

You might end up invoking MPX from the incorrect working directory from the shell (example: typing “`./mpx.exe`” or “`./mpx/mpx.exe`”), or by using a mis-configured shortcut.

If MPX is not started with the correct working directory path, you may encounter errors such as:

- The `help` command claiming that there is no help for any commands you try, even valid ones.
- The `ls` command returning an error even when given with no arguments.

The MPX Shell

MPX implements a command-line user interface (CLUI) or "shell" similar to that of MS-DOS or Unix. You type commands (some of which accept arguments), and the results are printed to your terminal.

MPX displays a prompt when it is ready to accept commands; by default the prompt is "MPX\$".

In the following example command line, "date" is the MPX shell command to run, "2008" is the first argument to that command, "12" is the second argument, and "30" is the third argument:

```
MPX$ date 2008 12 30
```

Not all commands accept arguments. Read below to learn what commands are available, and pay particular attention to the `help` command.

MPX Shell Commands

These commands are available in version R1 of MPX:

- `help`
- `commands`
- `exit`
- `version`
- `date`
- `ls`

Each of the commands are documented in-detail in the following sections.

The “help” Command

`help` provides online help for users of MPX. This command is useful if you are forced to use MPX without access to this document (“Paul Prince’s Multi-Programming Executive: User’s Manual”). It can also be handy when you have simply forgotten the name of a command, or the order of a command’s arguments, et cetera.

Usage

This command has two different forms, depending on the number of arguments given:

With no arguments

If no arguments are given, `help` prints a list of valid commands.

With 1 argument

If the name of a valid command is given as an argument, `help` shows the detailed description for that command.

Errors

If the wrong number of arguments is given, an error saying so will be displayed.

If a single argument is given but it doesn’t match a valid MPX shell command, you will be told that there is no help available for that command name.

If all commands, even valid ones, return the “no help available” response, check to see if the `help/` subdirectory has been removed or corrupted; MPX stores its online help data there. You may also see this happen if you run MPX from a working directory different from the one that contains the `mpx.exe` file.

Examples

The following lines demonstrate valid usages of this command:

- `help`
- `help help`
- `help version`

The “commands” Command

`commands` displays a list of all valid MPX shell commands. This is very similar to calling `help` with no arguments.

Deprecation Warning

This command has been slated for removal in a future version of MPX; you should use the command `help` instead!

Usage

This command takes no arguments.

Errors

This command produces no error messages.

Examples

Since this command takes no arguments, only the following line represents valid usage:

- `commands`

The "exit" Command

`exit` is used to terminate MPX and return to the host environment. After running `exit`, the user will be prompted "Are you sure you want to terminate MPX? [y/n]". In order to actually exit MPX, you must respond to this prompt by typing "y" or "yes" (not case-sensitive) and pressing return. Any other input will cancel the `exit` command and return to the MPX shell prompt.

Usage

This command takes no arguments.

Errors

This command is never expected to produce and errors during normal conditions.

Examples

Since this command takes no arguments, only the following line represents valid usage:

- `exit`

The “version” Command

`version` displays the version of the MPX software that is running (R1, R2, etc.).

The date when the current version was officially submitted for evaluation is also printed.

Usage

This command takes no arguments.

Errors

This command produces no error messages.

Examples

Since this command takes no arguments, only the following line represents valid usage:

- `version`

The "date" Command

date is used to both view and change the MPX system date.

On startup, MPX initializes its internal date from the host environment. You may change the MPX system date at any time by calling the **date** command with three arguments: the new year, month, and day (in that order).

Note that the date is never transmitted back to the host environment from MPX, and changes to the MPX system date will not be saved between invocations of MPX.

Usage

This command has two different forms, depending on the number of arguments given:

With no arguments

If no arguments are given, **date** prints the current MPX system date.

With 3 arguments

If you call **date** with three arguments representing a valid date (in order: the year, the month, and the day), the MPX system date will be changed to that date.

Errors

The user will be notified if any of the following occur: An invalid date was specified by the user, the wrong number of arguments were given to **date**, or an undefined error was encountered in trying to set the date.

Examples

The following lines demonstrate some valid usages of this command:

- **date**
- **date 2010 2 28**
- **date 2011 03 15**

The "ls" Command

`ls` is used to list MPX files in a given directory. MPX files are executable files suitable for running under MPX.

Called without arguments, `ls` will list the MPX files from the default MPX file directory ("`./mpx/`").

Otherwise, specify the path to the desired directory as the first and only argument to `ls`.

MPX does not verify that listed files are valid MPX files, but instead assumes that any file whose name ends in "`.MPX`" (not case-sensitive) is a valid MPX file.

Usage

This command has two different forms, depending on the number of arguments given:

With no arguments

If no arguments are given, `ls` lists the MPX files from the default MPX file directory.

With 1 argument

If you call `ls` with one argument representing a valid directory path, then the MPX files in that directory will be listed.

Errors

The user will be notified if any of the following occur: An invalid directory was specified by the user, the wrong number of arguments were given to `ls`, or an undefined error was encountered in trying to access the specified directory.

If you receive the "failed to open directory" error even when calling `ls` without arguments, check to see that the "`./mpx/`" directory exists and is not corrupt, that you have appropriate permissions to access it and the files therein, and that you are invoking `mpx.exe` from the correct working directory.

Examples

The following lines demonstrate some valid usages of this command:

- `ls`
- `ls ./somedir`
- `ls C:\PROGRA~1\PATH\TO\A\DIR\`