TechOS

Programmer Manual

By:

Ben Culkin

Jared Miller

Lucas Darnell

Table of Contents

Page 1

* Overview
* File Summaries

Page 2

* File Summaries continued

Page 3

* Global variables
* Cross Reference

Page 4

* Index

Overview

Our TechOS is a command line OS that is being made so we can get a good perspective on how Operating Systems are made and operate. The current state is very simple, with this only being the first module, and has only a few commands implemented. The project is written in C and our group is using Git to keep track of files.

For this first module, it involves only the command handler and the basic looping structure for the commands. The commands that can currently be used are datefmt, exit, help, setdate, version, and date.

File Summaries

**commands.c:** Contains all the code to handle commands when they are used

**commands.h:** Header file used by commands.c for method prototyping, struct declaring, constant variable declaration, and prototyping the command methods with a macro called DELCOM.

**techos.c:** The main file that handles running the OS.

**techos.h:** Header file for use in techos.c that has prototyping, a file declared, and some constant variables.

**date.1:** Manual file for the date command

**datefmt.1:** Manual file for the datefmt command

**exit.1:** Manual file for the exit command

**help.1:** Manual file for the help command

**setdate.1:** Manual file for the setdate command

**version.1:** Manual file for the version command

Data Structures

**Struct command**:

**Usage:** Contains all the information necessary to register a command into TechOS

**Attributes:**

* + - **Const char \*name:** The name of the command
    - **Const char \*brief:** A brief description of the command. Used in command listing
    - **Int (\*comfun)(int, char \*\*, char \*):** The handler for the command. The first argument is the number of arguments, the second is an array of the arguments that were separated by spaces, and the third is the unsplit command line. It should return a status code with 0 indicating success, positive indicating non-fatal error, and negative indicating fatal error.

**Int main():**

**Usage:** The main function of the program. Does setup for commands, runs the command handler, then tears down the commands.

**Void comhan():**

**Usage:** The command handler loop. Reads commands from the user, then parses and executes them.

**Struct command parsecom(char \*name):**

**Usage:** Create/Find an appropriate instance of **struct command** for the given command. May be the INVALID\_COMMAND if no command with that name exists.

**int execcom(struct command com, char \*argmarker, char \*argline):**

**Usage:** Execute the indicated command with the indicated arguments.

**Void initcoms():**

**Usage:** Do setup necessary for commands to work.

**Void disposecoms():**

**Usage:** Cleanup everything that initcoms() did.

**HANDLECOM(name):**

**Usage:** Macro for declaring command handlers. HANDLECOM(foo) gives the following header ‘int handle\_foo(int argc, char \*\*argv, char \*argl)’. Argc is the number of CLI arguments, argv is the arguments broken up by spaces, and argl is the unbroken arguments.

**HANDLECOM(exit):**

**Usage:** Exits TechOS, after confirmation.

**HANDLECOM(version):**

**Usage:** Prints the version/author information.

**HANDLECOM(date):**

**Usage:** Print the current date/time in the current format.

**HANDLECOM(datefmt):**

**Usage:** Set the format used for date input/output.

**HANDLECOM(setdate):**

**Usage:** Set the current date.

**HANDLECOM(help):**

**Usage:** Either list commands, or pull up the manual for a command.

Global Variables

**static const int major\_ver**

**static const int minor\_ver**

These variables are used to keep track of the current version

**static const int NUM\_COMMANDS**

Keeps track of the number of commands

**static const int MAX\_ARG\_COUNT**

Limits the amount of arguments a command can take

**static char \*in\_datefmt**

**static char \*out\_datefmt**

These variables take care of holding the date format and helping with the changes that happen to the date format.

**Static FILE \*strem**

The current stream to read commands from.

Cross Reference

**Function:** main()

**Calls:** initcoms(), comhan(), disposecoms();

**Called By:** nothing

**Function:** comhan()

**Calls:** parsecom(), execcom()

**Called By:** comhan()

**Function:** parsecom()

**Calls:** nothing

**Called By:** comhan()

**Function:** execcom()

**Calls:** all the handle\_\* functions through a func. pointer

**Called By:** comhan()

**Function:** initcoms()

**Calls:** nothing

**Called By:** main()

**Function:** disposecoms()

**Calls:** nothing

**Called By:** main()

**Function:** handle\_version()

**Calls:** nothing

**Called By:** execcom() through com.comfun

**Function:** handle\_date()

**Calls:** nothing

**Called By:** execcom() through com.comfun

**Function:** handle\_datefmt()

**Calls:** nothing

**Called By:** execcom() through com.comfun

**Function:** handle\_setdate()

**Calls:** nothing

**Called By:** execcom() through com.comfun

**Function:** handle\_help()

**Calls:** nothing

**Called By:** execcom() through com.comfun

**Function:** handle\_exit()

**Calls:** nothing

**Called By:** execcom() through com.comfun

Index