Kernel- Project

CMSC 125 Course Project

User Manual

Author: Ricafelle Dorothy Z. Fortes

Version: 1.0

TABLE OF CONTENTS

Introduction

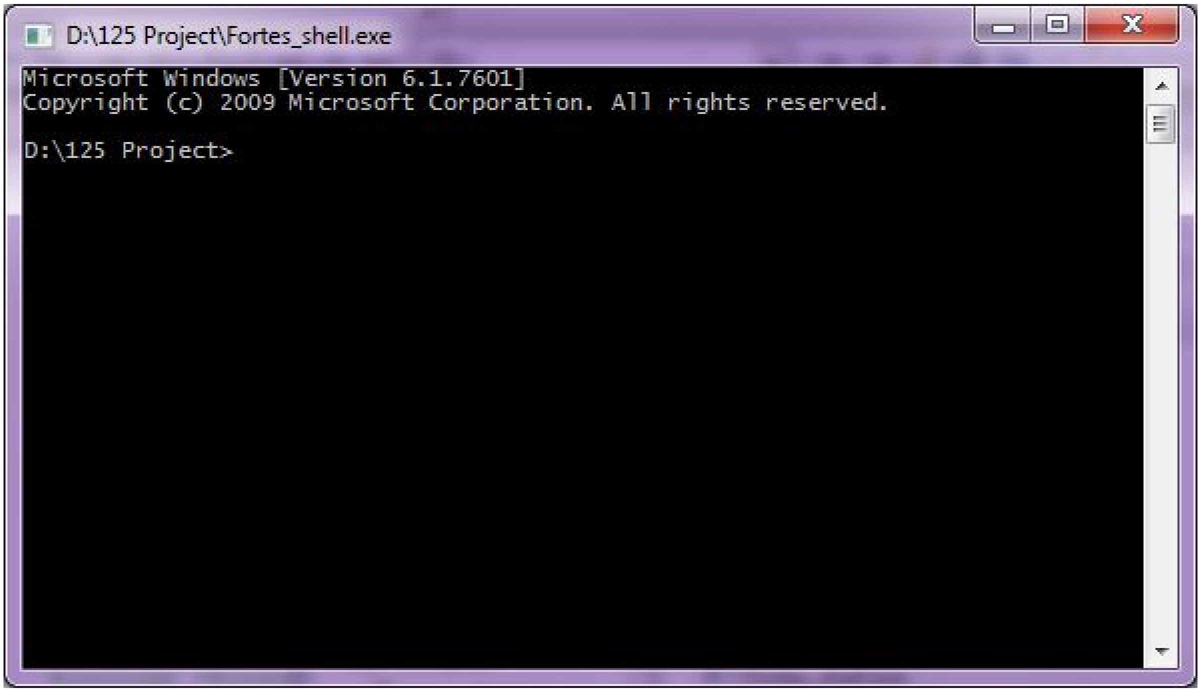
This project is the final requirement for the course CMSC 125 Operating Systems, under Mrs. Paula Esplanada-Mayol. The Kernel Project is the implementation of kernel interface that we see in our operating systems. In a Linux OS, we call it Terminal; and in a Windows OS, we call it Command Prompt. This shell is used to manipulate files and other services inside the OS. The Kernel Project is implemented using C language. In this tutorial, we will cover the basic commands and usage of the Kernel Project so that you will feel comfortable in using this resource.

Using the exe file

To run the program and start the kernel shell, you must do the following simple steps:

Step 1: Locate the exe file. In this shell, the exe file is named as **Fortes_shell.exe**
Step 2: Double click the exe file, or right click the exe file and select **Open**.

After following these steps, you will be presented with a window that look similar to Figure 1 below.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\125 Project>
```

Figure 1. Fortes_shell prompt

The Fortes_shell is simply a window like command prompt that, by default, displays the current directory or folder that you are in. The window also has a blinking cursor ready for you to type your commands. For example, in Figure 1 above, you can see that it says D:\125 Project>. The D:\125 Project> is the prompt and it tell me that I am currently in the d:\125 project directory. If I was in the directory c:\users, the prompt would instead look like this: C:\Users>. The shell also displays the version of your OS and the year.

To use the shell, simply type the command and instructions you want and then press enter. In the next section, we will discuss some useful commands and how to see all available built in commands for the shell prompt.

Useful commands

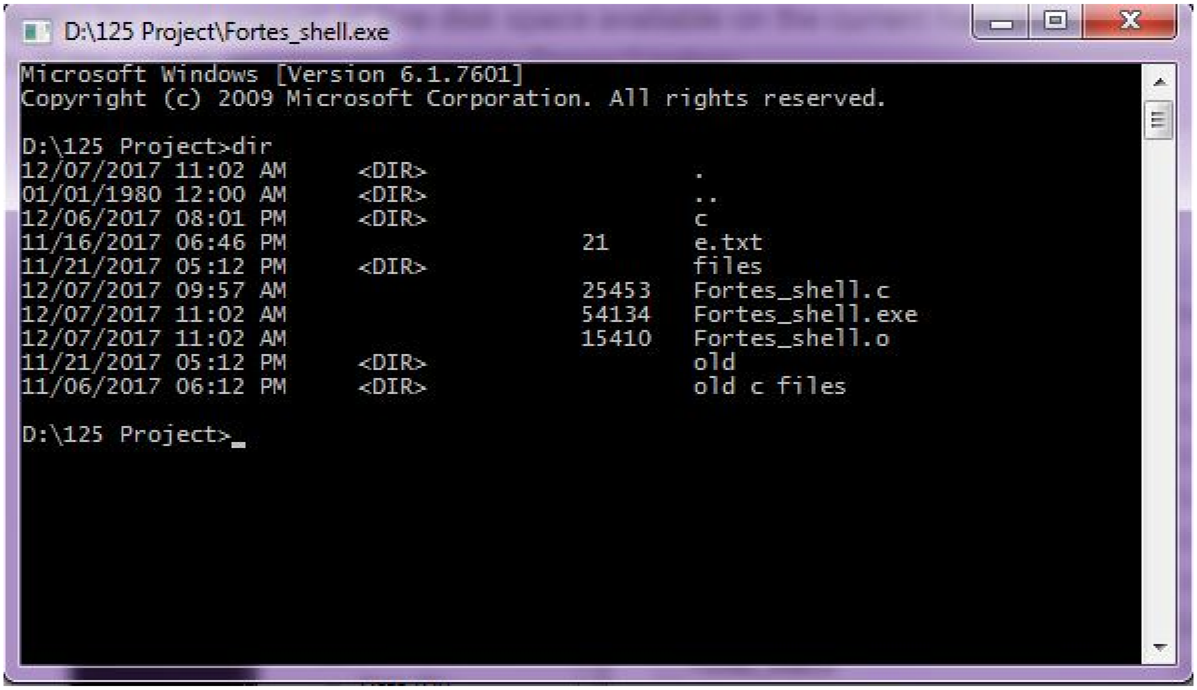
The shell program has built in commands that are very useful like what we see in our command prompts or terminals. I have outlined below some of the more important commands and further instruction on how to find information on all the available commands.

The **Help** command - This command will list all the current available commands that are built into the shell prompt.

The **Exit** command - This command will close the shell prompt. Simply type **exit** and press enter and the shell prompt will close.

The **CD** command - This command allows you to change your current directory or see what directory you are currently in. To use the CD command, you would type **cd ‘directoryname’** and press enter.

The **DIR** command - If used without an argument, this command will list the files and directories contained in your current directory. To use the command, you would just type **dir** and press enter and you will see a listing of the current files in the directory you are in, including information about their file size, date and time they were last written to.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\125 Project>dir
12/07/2017 11:02 AM     <DIR>                   .
01/01/1980 12:00 AM     <DIR>                   ..
12/06/2017 08:01 PM     <DIR>                   c
11/16/2017 06:46 PM                    21       e.txt
11/21/2017 05:12 PM     <DIR>                   files
12/07/2017 09:57 AM                    25453    Fortes_shell.c
12/07/2017 11:02 AM                    54134    Fortes_shell.exe
12/07/2017 11:02 AM                    15410    Fortes_shell.o
11/21/2017 05:12 PM     <DIR>                   old
11/06/2017 06:12 PM     <DIR>                   old c files

D:\125 Project>_
```

Figure 2. DIR

If you examine the screen above, you will see a listing of the directory. The first 2 columns are the date and time of the last write or modification to that file. Followed by whether or not tha particular entry is a directory or a file, then the size of the file, and finally, the name of the file. You may have noticed that there are two directories name **.** and **..**, which have special meaning in operating systems.

The **copy** command - This command allows you to copy files from one location to another. To use this command, you would type **copy 'thefiletocopy' 'wheretocopy'**. For example if you have the file d:\125 Project\test.txt and would like to copy it to c:\users\test.txt you would type **copy d:\125 Project\test.txt c:\users\test.txt** and press enter. If the copy is successful it will tell you so and give you back the prompt. If you are copying within the same directory you do not have to use the path. Here are some examples and what they would do:

copy test.txt test.bak	Copies the test.txt file to a new file called test.bak in the same directory

copy test.txt \windows	Copies the test.txt file to the \windows directory.

At this point, the help command will help you learn about the other available commands that was made during the implementation of the project.

Conclusion

Hopefully this brief introduction into the kernel project and its implementation in c will help you understand more about our operating systems.