[2019 Network System Programming Homework 4]

This homework focuses on system programming and pipe.

Part 1:

1. Write, compile, and run a program named **hostinfo** that prints out system information in the following format.
   Sample output:

```
ubuntupc@ubuntu:~/Desktop/D083040001/part1$ ./hostinfo
hostname: ubuntu
5.0.0-23-generic
hostid: 8323329
```

2. Write, compile, and run a program named **mydate** that prints out the day and time in the following format.
   Sample output:

```
ubuntupc@ubuntu:~/Desktop/D083040001/part1$ ./mydate
Oct 8(Tue), 2019 1:22 PM
```

3. Write a program called **printdir** that prints the current directory. Determine what size buffer to pass to **getcwd()** for dynamic allocation.(**Do not** use pwd().)
   Sample output:

```
ubuntupc@ubuntu:~/Desktop/D083040001/part1$ ./printdir
/home/ubuntupc/Desktop/D083040001/part1
```

4. Write a program called **mycat** that is a simple version of the program cat. The program takes exactly one file name as argument; you should open it for reading and display its contents to the screen. Check that there is exactly one argument (argc == 2) and if not, display the usage message "Usage: mycat filename".
   Sample output:

```
ubuntupc@ubuntu:~/Desktop/D083040001/part1$ cat 123
123456
ABCDE
***
ubuntupc@ubuntu:~/Desktop/D083040001/part1$ ./mycat 123
123456
ABCDE
***
```

5. Write **pipe_ls** to practice using **pipe()** and **dup()**. Have your process start ls (using **fork()** and **exec()**) but read the output from ls over a pipe. The ls program writes output on descriptor 1, so some work has to be done to get the pipe connected. Write what you read on the pipe to **stdout**.
   Sample output:

```
ubuntupc@ubuntu:~/Desktop/D083040001/part1$ ./pipe_ls
total 112
-rwxrw-rw- 1 ubuntupc ubuntupc   17 Oct  4  2018 123
-rwxrw-rw- 1 ubuntupc ubuntupc 8520 Oct  4  2018 hostinfo
-rwxrw-rw- 1 ubuntupc ubuntupc  477 Oct  4  2018 hostinfo.c
-rwxrw-rw- 1 ubuntupc ubuntupc 2192 Oct  4  2018 hostinfo.o
-rwxrw-rw- 1 ubuntupc ubuntupc  549 Oct 20  2017 makefile
-rwxrw-rw- 1 ubuntupc ubuntupc 8640 Oct  4  2018 mycat
-rwxrw-rw- 1 ubuntupc ubuntupc  550 Oct  4  2018 mycat.c
-rwxrw-rw- 1 ubuntupc ubuntupc 2296 Oct  4  2018 mycat.o
-rwxrw-rw- 1 ubuntupc ubuntupc 8424 Oct  4  2018 mydate
-rwxrw-rw- 1 ubuntupc ubuntupc 1639 Oct  4  2018 mydate.c
-rwxrw-rw- 1 ubuntupc ubuntupc 4128 Oct  4  2018 mydate.o
-rwxrw-rw- 1 ubuntupc ubuntupc 8528 Oct  4  2018 pipe_ls
-rwxrw-rw- 1 ubuntupc ubuntupc  409 Oct  4  2018 pipe_ls.c
-rwxrw-rw- 1 ubuntupc ubuntupc 2128 Oct  4  2018 pipe_ls.o
-rwxrw-rw- 1 ubuntupc ubuntupc 8568 Oct  4  2018 printdir
-rwxrw-rw- 1 ubuntupc ubuntupc  386 Oct  4  2018 printdir.c
-rwxrw-rw- 1 ubuntupc ubuntupc 2024 Oct  4  2018 printdir.o
```

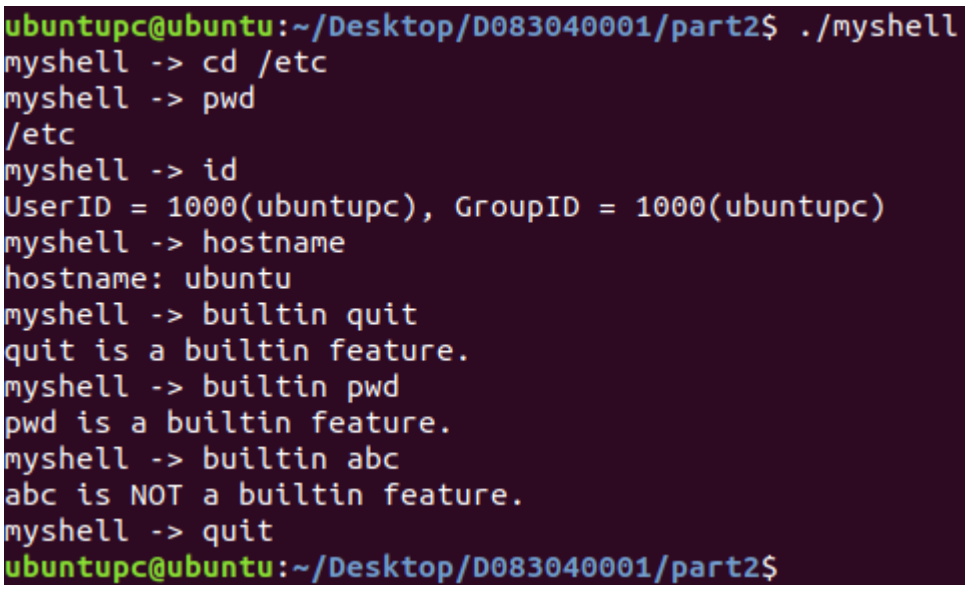
Part 2:

1. Edit the **builtin.c** file to recognize **cd**, **pwd**, **id**, **hostname** and **builtin**. Write functions implementing these commands, and compile then into your shell. The **builtin** command lists the functions built into your shell.

   Files provided:

   builtin.c, parse.c, shell.c, shell.h

   Sample output:

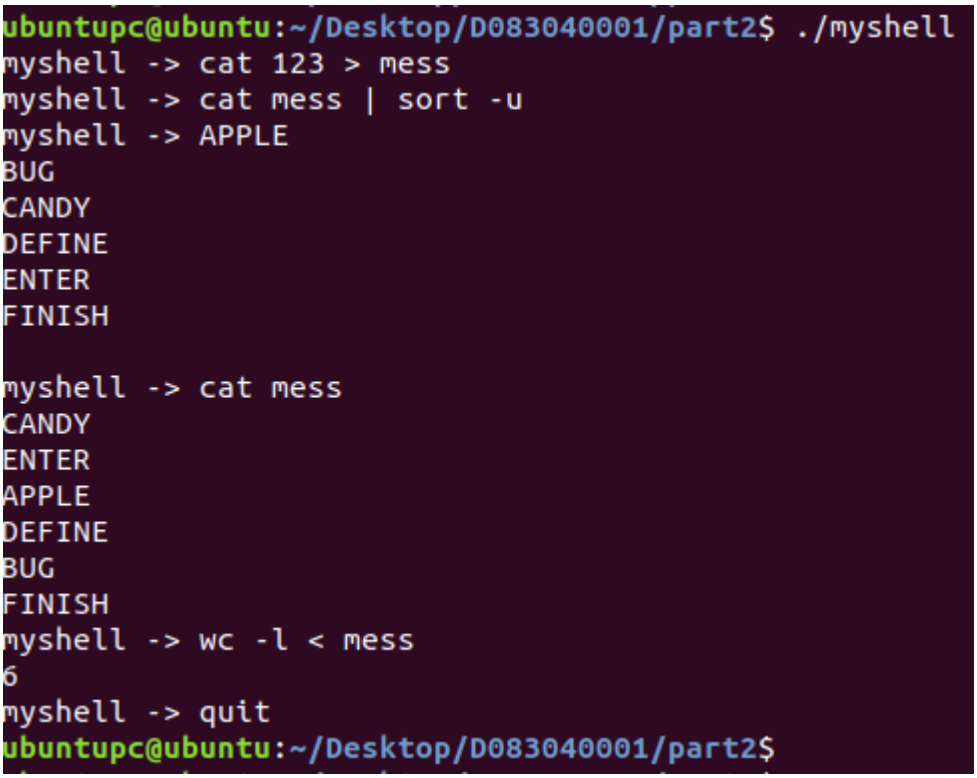


2. Modify the **redirect_in.c** file to recognize standard input and **redirect_out.c** file to recognize standard output. Add code to the **pipe_present.c** file to check for the pipe symbol. Add code to the **pipe_command.c** file to create a process to execute each of the pipe. Modify **is_background.c** to check the "&" symbol. Alter the **run_command.c** file to call these functions.

   Files provided:

   redirect_in.c, redirect_out.c, run_command.c, pipe_present.c, pipe_command.c, is_background.c

   Sample output:



Part 3:

1. The first look up project lab familiarizes you with the format of the dictionary by completing the **convert.c** program that creates the dictionary of fixed-length records (fixrec) from a file of variable-length entries (dict). Add code to **convert.c** to change an editable file into a fixed-length record format.

   File provided:

   convert.c, dict.h, dict

   Sample output:

```
ubuntupc@ubuntu:~/Desktop/D083040001/part3$ ./convert dict myfixrec
ubuntupc@ubuntu:~/Desktop/D083040001/part3$ ls -l dict myfixrec
-rwxrw-rw- 1 ubuntupc ubuntupc  1920 Oct  3  2016 dict
-rw-r--r-- 1 ubuntupc ubuntupc 10240 Oct  8 06:41 myfixrec
ubuntupc@ubuntu:~/Desktop/D083040001/part3$
```

2. Add code to the **lookup1.c** file to do a simple linear search through a file of fixed length records. Link with **main.c**, the user interface module.

   File provided:

   lookup1.c, main.c

   Sample output:

```
ubuntupc@ubuntu:~/Desktop/D083040001/part3$ ./file_lookup myfixrec
What word do you want : work
work : The curse of the drinking classes.
What word do you want : cynic
cynic : A blackguard who sees things as they are and not as they ought to be.
What word do you want : beauty
beauty : The power by which a woman charms a lover and terrifies a husband.
What word do you want : homework
homework : Not Found!
What word do you want : ^C
ubuntupc@ubuntu:~/Desktop/D083040001/part3$
```

Rules:

1. Please use C language in this homework and run your program on Ubuntu 18.04.
2. Please provide Makefile to compile your homework; otherwise, you will get ZERO.
3. Do not copy others homework.
4. If you have any question, please send email to sp_ta@net.nsysu.edu.tw or come to F5018, but TA does not help to debug.

Upload:

1. Please compress your homework into zip or tar archive.
2. Upload your homework to **NSYSU Cyber University**.
3. Naming rules:"**StudentID_SP_HW4.zip**".
   For example: D083040001_SP_HW4.zip
4. Deadline: 2019/10/15 23:59