【2021 Network System Programming Homework 2】

1. Please use C language for this homework and make sure it can run correctly on Ubuntu 20.04.
2. Please provide Makefile to compile your homework.
3. Do not copy the others homework definitely.
4. If you have any question, please send email to sp_ta@net.nsysu.edu.tw or drop by Room EC5018. However, TA will not help you to debug program.
5. Please write your homework with functions that have appeared in course slides (within course progress), or the textbook file provided in the homework file. This homework is meant to let you practice system functions, not simply write a program that meets the requirement. Only if the requirement can't be done by those system functions, you are allowed to use other functions.

Turn in your homework:

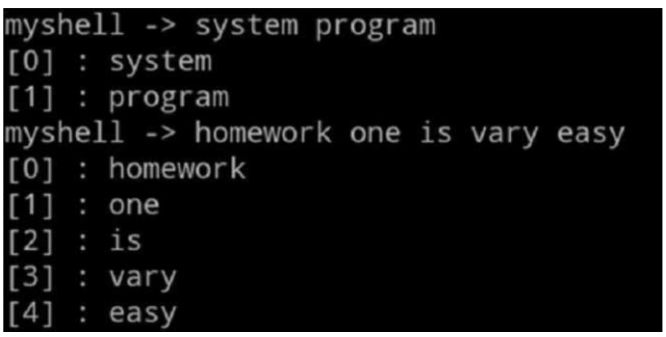
1. Please compress your homework into zip file.
2. Naming rules: "SP_HW2.zip".
3. Upload your homework (zip file) to NSYSU Cyber University (網路大學).
4. Deadline: 2021/10/19 14:10. You cannot get any credit if you do not turn in your homework before the deadline.

Homework description:

Part 1

1. Edit the parse.c file to use *strtok()* and *realloc( )* to implement the *parse()* and *free_argv()* functions.
2. Files provided:

   shell.h shell.c parse.c
3. Example:

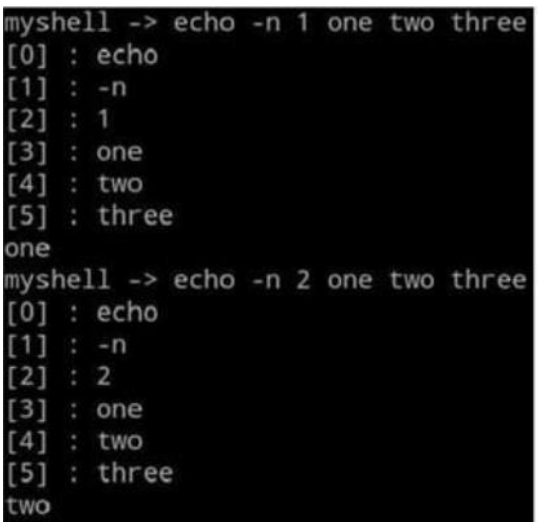


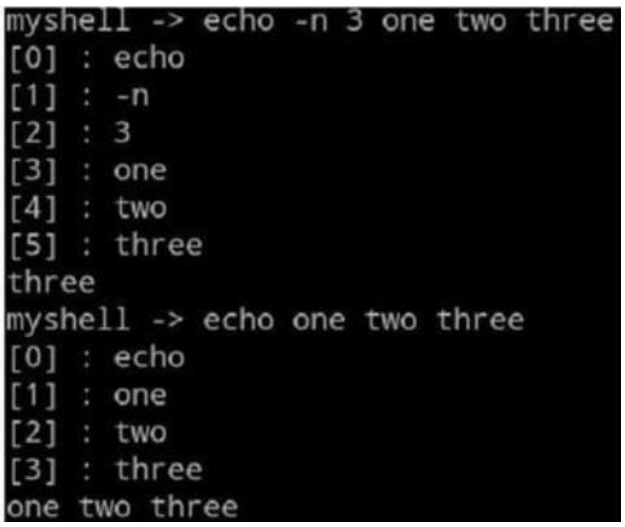
Part 2

1. Add code to the builtin.c stub to recognize the echo, quit, exit, logout and bye commands. Write functions implementing these commands, and add a new line for each command to table inbuilts[] just above the line {NULL, NULL}.
2. File provided:

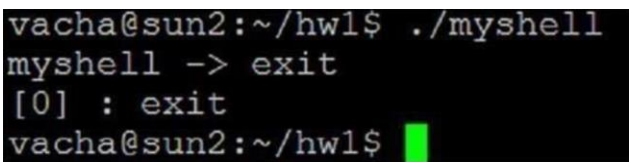
   Builtin.c
3. echo example:

   echo print all strings echo -n N: print the specified string





4. quit example: exit, quit, logout and bye terminate the program.



Part 3

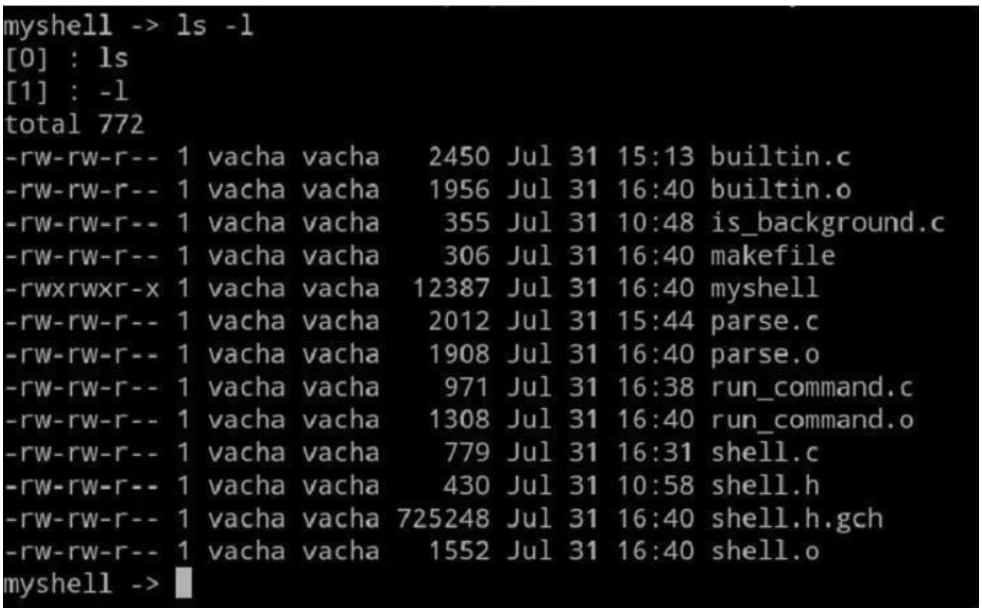
1. Edit the run_command.c file so a child process is created to run the command, and the parent waits for the child process to terminate. Check for builtin commands first, create a new process only for commands which are not built in. Use the parser from pervious labs to create from the command line the argv array passed to the child.

   Hint: You can use *exec()/execvp()* function, *waitpid()* function and the *fork()* system call.

2. File provided:

   run_command.c

3. Example:



Part 4

1. Edit the is_background.c file to detect an "**&**". Alter the run_command.c file so that if a task is running in the background, the parent does not wait. Do not worry about background processes becoming zombies at this point; this will be addressed later.
2. File provided:

   is_background.c
3. Example: