Programming Assignment #1

Due : Sunday, October 27, 2024), 11:59 PM

1. Introduction

In this assignment, you will implement six simple commands in a Linux mini shell.

2. Problem specification

The Objective of PA1 is to implement six Linux commands in C.

The function specification of command is explained in the below. In general, the execution result of commands is required to be the same as the output normally given by the same commands in Linux. You can execute each command in your Linux environment to confirm its output.

In this assignment package, the file *pa1_skeleton.c file* is provided, and you need to implement the functions in that file such as *ls, head, tail, mv, cp,* and *pwd* to complete the corresponding commands.

2.1 ls

```
>> ls dir_path
>> ls dir_path -al
```

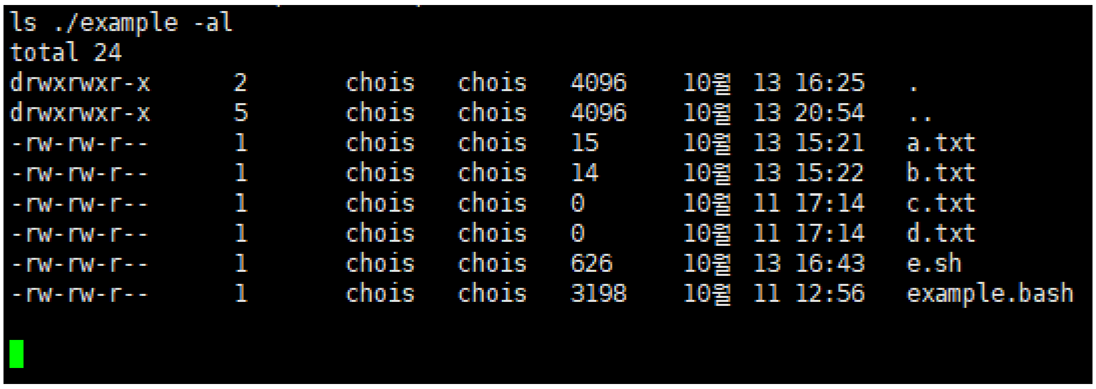
Function specification:

```
int ls(char *dir_path, char *option){
}
```

If the function is executed normally, 0 is returned, otherwise, -1 is returned. If the *-al* option is entered, the option argument is *"-al"*. Otherwise, NULL.
'-al' option should print the output in the following.

```
Permissions / Number of hard links / File owner / File group / File size /
Last modified time / File name
```

The output files should be the same as when you put *ls -al* command in Linux.



```
ls ./example -al
total 24
drwxrwxr-x      2       chois   chois   4096    10월  13 16:25   .
drwxrwxr-x      5       chois   chois   4096    10월  13 20:54   ..
-rw-rw-r--      1       chois   chois   15      10월  13 15:21   a.txt
-rw-rw-r--      1       chois   chois   14      10월  13 15:22   b.txt
-rw-rw-r--      1       chois   chois   0       10월  11 17:14   c.txt
-rw-rw-r--      1       chois   chois   0       10월  11 17:14   d.txt
-rw-rw-r--      1       chois   chois   626     10월  13 16:43   e.sh
-rw-rw-r--      1       chois   chois   3198    10월  11 12:56   example.bash
```

2.2 head

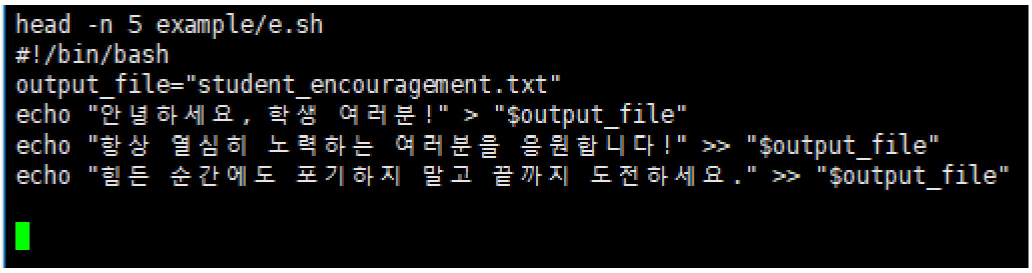
```
>> head -n K file_path
```

Function specification:

```
int head(char *file_path, char *line){

}
```

If the function is executed normally, 0 is returned, otherwise, -1 is returned. In the **line** argument, (*K* >0) on the command is input.

The result of using the head command on example/e.sh, is as follows.



```
head -n 5 example/e.sh
#!/bin/bash
output_file="student_encouragement.txt"
echo "안녕하세요, 학생 여러분!" > "$output_file"
echo "항상 열심히 노력하는 여러분을 응원합니다!" >> "$output_file"
echo "힘든 순간에도 포기하지 말고 끝까지 도전하세요." >> "$output_file"
```

2.3 tail

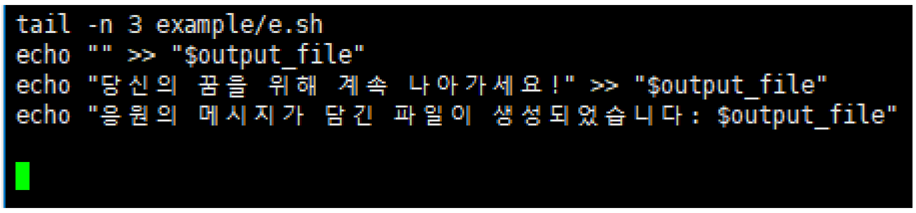
```
>> tail -n K file_path
```

Function specification:

```
int tail(char *file_path, char *line){

}
```

If the function is executed normally, 0 is returned, otherwise, -1 is returned. In the **line** argument, (*K* >0) on the command is input.

The result of using the tail command on example/e.sh, is as follows.



```
tail -n 3 example/e.sh
echo "" >> "$output_file"
echo "당신의 꿈을 위해 계속 나아가세요!" >> "$output_file"
echo "응원의 메시지가 담긴 파일이 생성되었습니다: $output_file"
```

2.4 mv & cp

```
>> mv file_path1 file_path2
```

Function specification:

```
int mv(char *file_path1, char *file_path2){

}
```

If the function is executed normally, 0 is returned, otherwise, -1 is returned.

```
>> cp file_path1 file_path2
```

Function specification:

```
int cp(char *file_path1, char *file_path2){

}
```

If the function is executed normally, 0 is returned, otherwise, -1 is returned.

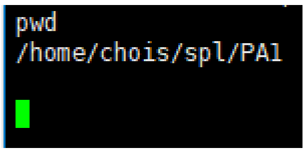
2.5 pwd

```
>> pwd
```

Function specification:

```
int pwd(){

}
```

If the function is executed normally, it prints the current working directory path.



```
pwd
/home/chois/spl/PA1
```

3. Restriction

- The length of file path will be no larger than 200 bytes.
- If the argument name is *file_path*, it is a valid path only if it is a path to file. For *dir_path*, it is a valid path only if it is a path to directory.
- *Head, tail, mv and cp* commands will not be tested with any directory in the path. (If a directory is given in the path for those commands, then simply print 'Error: invalid path'.). If the path is invalid, print the following.

```
>> ERROR: invalid path
```

- If the command is executed abnormally, this statement should be printed as follows.

```
>> ERROR: The command is executed abnormally
```

This has implemented on the template so that it is automatically output when the function returns -1.

- exec functions(such as execv(), execl()) and system functions are forbidden.
- For file I/O, using only Unix I/O functions are allowed.
- **You are not allowed to use any header files other than those already included in the skeleton. If you need to use a different function, create and implement it as done with the 'stringcmp' function.**
- **The program terminates with the 'quit' command; this functionality is already**

implemented in the skeleton, so do not modify it.

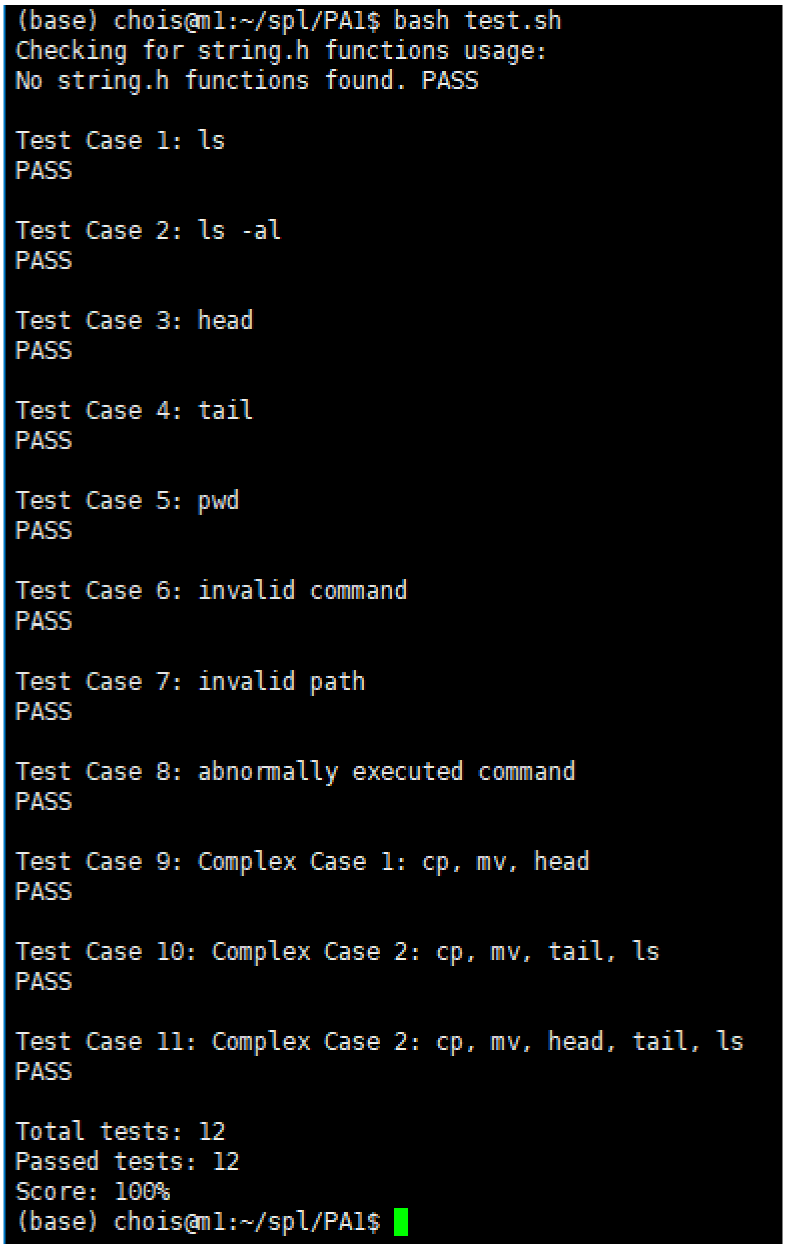
- **Do not modify the *test.sh* and files in *testcase/ output/ example/*.**

4. Tips

- You can use the *test.sh* with predefined testcases. Just execute the test.sh

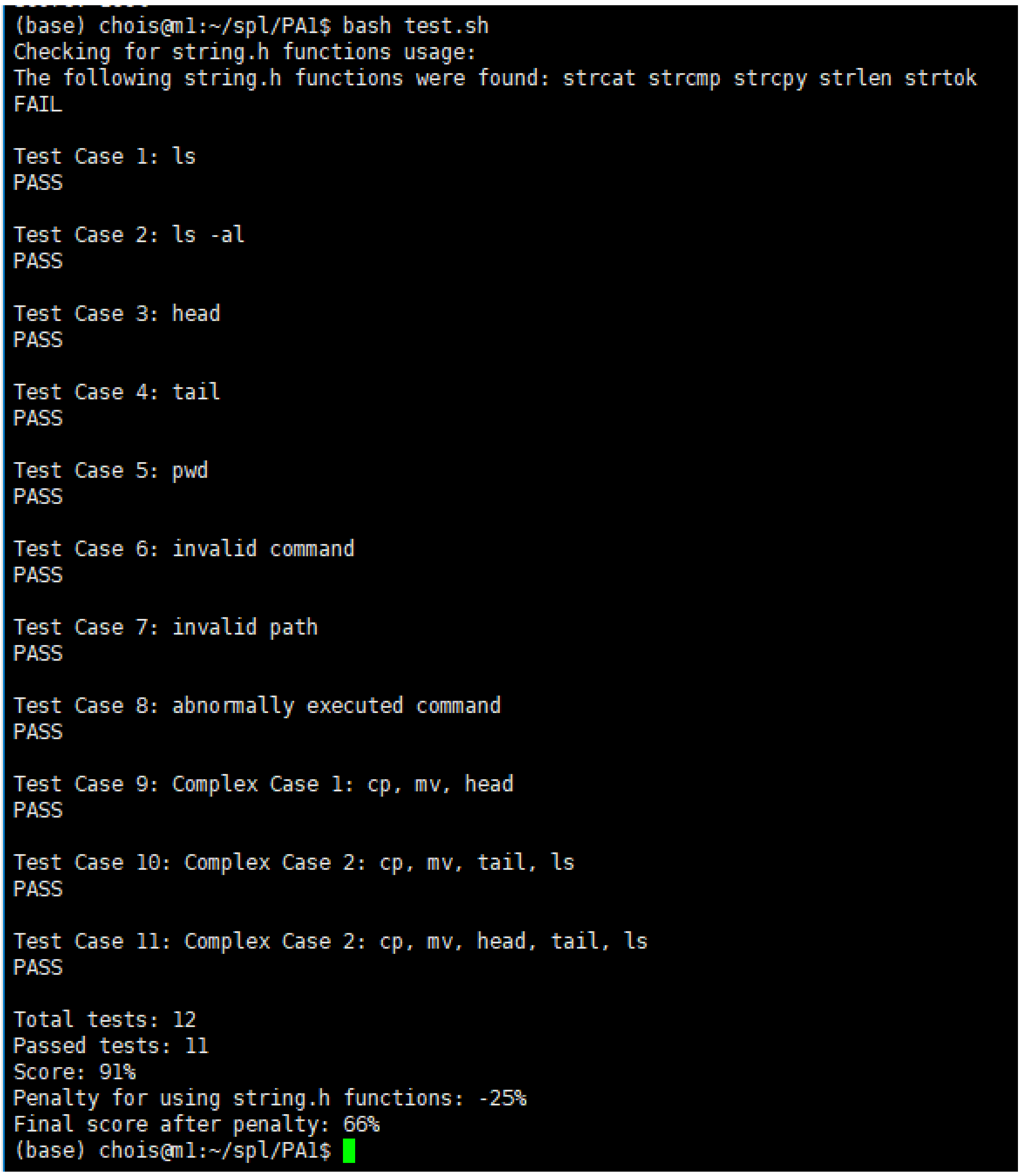
```
>> bash test.sh
```

- If your code meets the assignment requirements, you will see the following results.



```
(base) chois@m1:~/spl/PA1$ bash test.sh
Checking for string.h functions usage:
No string.h functions found. PASS

Test Case 1: ls
PASS

Test Case 2: ls -al
PASS

Test Case 3: head
PASS

Test Case 4: tail
PASS

Test Case 5: pwd
PASS

Test Case 6: invalid command
PASS

Test Case 7: invalid path
PASS

Test Case 8: abnormally executed command
PASS

Test Case 9: Complex Case 1: cp, mv, head
PASS

Test Case 10: Complex Case 2: cp, mv, tail, ls
PASS

Test Case 11: Complex Case 2: cp, mv, head, tail, ls
PASS

Total tests: 12
Passed tests: 12
Score: 100%
(base) chois@m1:~/spl/PA1$
```

- If your code fails to meet certain conditions, you will see the following results.



```
(base) chois@ml:~/spl/PA1$ bash test.sh
Checking for string.h functions usage:
The following string.h functions were found: strcat strcmp strcpy strlen strtok
FAIL

Test Case 1: ls
PASS

Test Case 2: ls -al
PASS

Test Case 3: head
PASS

Test Case 4: tail
PASS

Test Case 5: pwd
PASS

Test Case 6: invalid command
PASS

Test Case 7: invalid path
PASS

Test Case 8: abnormally executed command
PASS

Test Case 9: Complex Case 1: cp, mv, head
PASS

Test Case 10: Complex Case 2: cp, mv, tail, ls
PASS

Test Case 11: Complex Case 2: cp, mv, head, tail, ls
PASS

Total tests: 12
Passed tests: 11
Score: 91%
Penalty for using string.h functions: -25%
Final score after penalty: 66%
(base) chois@ml:~/spl/PA1$
```

- The actual evaluation may use test cases different from the ones provided.

5. Hand in instruction

- You should submit the codes with "student_id.tar.gz".

6. Logistics

- **Assignment should be done individually.**
- If the assignment is delayed, 10% will be deducted immediately after the deadline, and an additional 10% will be deducted every 24 hours. Delayed submissions will not be accepted

after 7 days of submission deadline.

- **Copying assignments will be punished in accordance with the lab's own regulations, and there will be significant disadvantages.**