

리눅스시스템 실습 #12

2021/12/08

2021-2

리눅스시스템(창병모 교수님)

TA: 오현주

[참고] 프로그램 13.5

```
ch13.c
~/ch13

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

/* 자식 프로세스를 생성하여 echo 명령어를 실행한다. */
int main() {
    int pid, child, status;
    printf("부모 프로세스 시작\n");
    pid = fork();

    if (pid != 0) {
        child = waitpid(pid, &status, 0); /* 부모 프로세스는 자식프로세스가 종료되기를 기다림 */
        printf("자식 프로세스 %d 끝\n", child);
        printf("부모 프로세스 끝\n");
    } else {
        execl("/bin/echo", "echo", "hello", NULL);
        fprintf(stderr, "첫 번째 실패");
        exit(1);
    }
}
```

```
sllab@localhost:~/ch13

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[sllab@localhost ch13]$ gcc -o ch13 ch13.c
[sllab@localhost ch13]$ ./ch13
부모 프로세스 시작
hello
자식 프로세스 13332 끝
부모 프로세스 끝
```

1. 쉘 인터프리터 작성

- 프로그램 13.5를 참고하여 다음과 같은 기능을 포함하는 쉘 인터프리터를 작성하시오. 프로그램 13.5는 무조건 echo 명령어를 실행하고 있다. 여기에서는 실행할 명령어를 다음과 같은 형태로 입력받는다.
 - **1) 명령어 전면 실행**
 - [shell] 명령어
 - **2) 명령어 후면 실행**
 - [shell] 명령어 &

1. 셸 인터프리터 작성

- 이 인터프리터를 구현하기 위해서 다음과 같이 진행한다.
 - 먼저 `strtok_r()` 함수와 `execvp()` 시스템 호출의 기능에 대해서 조사한다. (검색하기)
- 셸 인터프리터는 다음과 같이 동작한다.
 - 1) 프롬프트를 내주고 명령어를 입력으로 받는다.
 - 2) `strtok_r()` 함수를 이용하여 이를 명령줄 인수로 분리한다.
분리된 명령줄 인수를 다음 배열에 순차적으로 대입한다.

```
char *args[MAXARG];
```
 - 3) 자식 프로세스를 생성하여 자식 프로세스로 하여금 `execvp()` 시스템 호출을 이용하여 명령어를 실행하게 한다.
 - 4) 부모 프로세스는 자식 프로세스가 끝나기를 기다린다.
 - 5) (1)로 돌아가서 같은 과정을 반복한다.

1. 쉘 인터프리터 - 실행 결과

```
sllab@localhost:~/ch13
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[sllab@localhost ch13]$ ./shell
[shell] echo hello world!
hello world!
[shell] date
2019. 12. 02. (월) 01:12:52 KST
[shell] who
sllab      :0                2019-11-05 15:36 (:0)
sllab      pts/0            2019-12-02 01:07 (:0)
sllab      pts/1            2019-12-02 01:12 (:0)
[shell] ps
  PID TTY          TIME CMD
 1316 pts/1        00:00:00 ps
 32395 pts/1       00:00:00 bash
 32642 pts/1       00:00:00 shell
[shell] cat test.txt
안녕하세요.
이 파일은 test.txt입니다.
[shell] touch test.txt
[shell] touch a.c
[shell] rm a.c
[shell] mkdir hello
[shell] rmdir hello
[shell] cat -n test.txt
     1  안녕하세요.
     2  이 파일은 test.txt입니다.
[shell] cat -n
hello
     1  hello
world
     2  world
[shell] quit quit 입력 시 쉘 인터프리터 종료
[sllab@localhost ch13]$
```

- 그동안 사용한 명령어들이 정상적으로 작동해야 함.

```
sllab@localhost:~/ch13
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[sllab@localhost ch13]$ ./shell
[shell] ls -l
합계 20
-rwxrwxr-x. 1 sllab sllab 8888 12월  2 01:08 shell
-rw-r--r--. 1 sllab sllab 1496 11월 30 03:41 shell.c
-rw-rw-r--. 1 sllab sllab   50 12월  2 01:13 test.txt
[shell] ls
shell shell.c test.txt
[shell] █
```

1. 쉘 인터프리터 - 실행 결과

```
sllab@localhost:~/ch13
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[shell] sleep 10
█
```

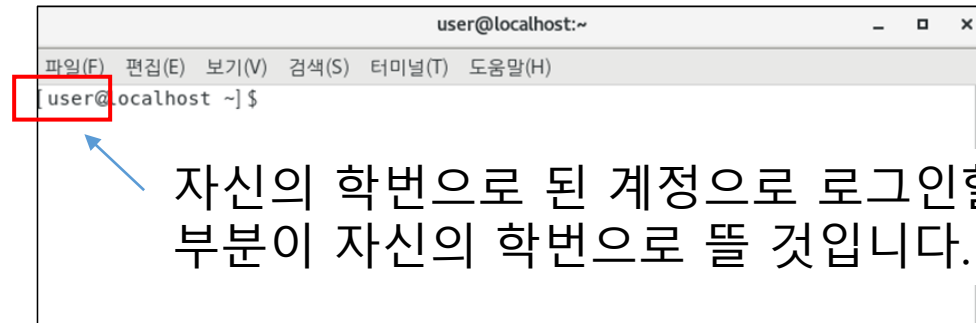
- 명령어 후면실행 시에는 다른 명령어 실행이 가능하다.

```
sllab@localhost:~/ch13
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[shell] sleep 10
[shell] sleep 10 &
[shell] date
2019. 12. 02. (월) 01:23:54 KST
[shell] ps
  PID TTY          TIME CMD
 10677 pts/0    00:00:00 sleep
 10869 pts/0    00:00:00 ps
 13857 pts/0    00:00:00 bash
 29500 pts/0    00:00:00 shell
```

두 개의 실행결과를 꼭 비교해서 보고서에 넣을 것

Lab12 과제 제출 – 보고서 + shell.c

- 각 문제별 **실행 화면 캡처 및 간략한 설명(2줄 이상)**을 보고서로 작성 ※ 압축해서 제출하기
 - 보고서에 실습번호, 분반, 과, 학번, 이름을 적을 것
 - 파일명: 실습번호_학번_이름(ex. Lab12_2031546_오현주)
 - 꼭 자신의 학번으로 된 계정으로 로그인하여 실습을 진행한 후 캡처를 하시기 바랍니다.

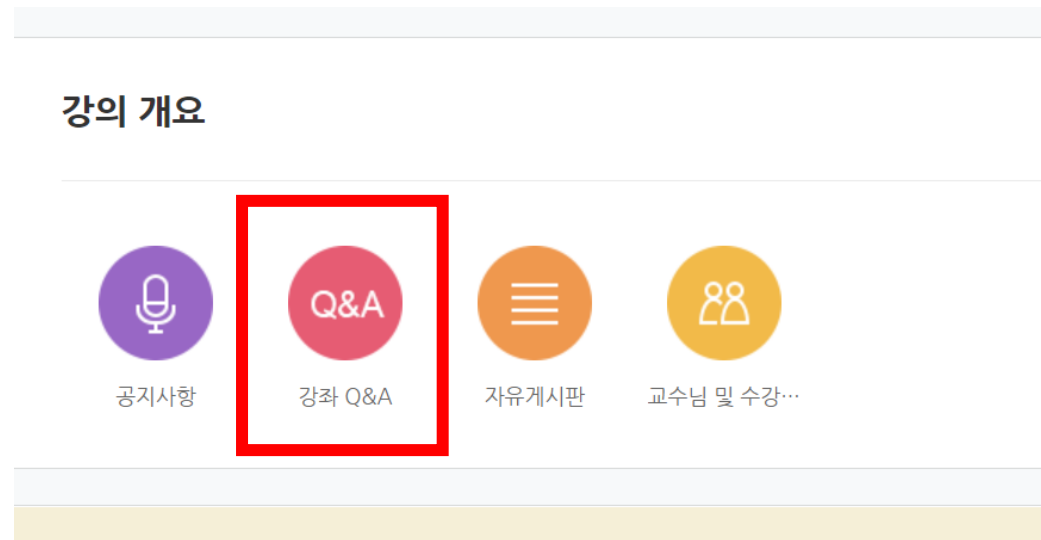


자신의 학번으로 된 계정으로 로그인할 경우 터미널의 이 부분이 자신의 학번으로 뜰 것입니다.

- 스노우보드 **Lab12**에 **실습번호_학번_이름**의 파일명으로 압축해서 제출할 것
 - 워드 혹은 한글 문서로 보고서를 작성하신 뒤 pdf로 변환하시기 바랍니다.
 - **shell.c** ※ c 코드 파일명 지키지 않을 경우 해당 문제 0점 처리됩니다.
- 위의 사항들을 지키지 않을 경우 **감점 요인이 되므로** 이 점 주의하시기 바랍니다.
- 기한: **12월 15일 수요일, 오후 11시 55분까지(과제 기한 체크 ★ ★)**
 - 늦게 제출할 경우(-2), 제출기한 이후 24시간까지만 받음.

실습 관련 문의

- 실습 관련 질문이 있는 경우 스노우보드 강좌 Q&A에 등록해주시기 바랍니다.



- 질문 작성 시, 내용을 구체적으로 명확하게 적어주시기 바랍니다.
- 코드를 첨부하고 싶을 경우, 캡처가 아닌 **텍스트 형식**으로 보내주세요.
 - 코드 파일도 함께 첨부해주세요