

Lab05

웹 인터프리터 작성

2021-1 시스템 프로그래밍(창병모 교수님)
TA: 오현주

실습 #1

간단한 쉘 인터프리터를 작성한다. 이 인터프리터는 Linux 명령어를 읽고 해석하여 실행시킨다. 이 인터프리터의 기능은 다음과 같다.

(1) 명령어 실행(Command execution)

```
[shell] cmd
```

(2) 후면 실행(Background execution)

```
[shell] cmd&
```

(3) 명령어 순차 실행

```
[shell] cmd1; cmd2; cmd3
```

(4) 입/출력 리디렉션(Input/Output redirection)

```
[shell] cmd > outfile
```

```
[shell] cmd < infile
```

실습 #1

- 프로그램 구성

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>
#define MAXARG 7

int main()
{
    char buf[256];
    char *args[MAXARG];
    char *s;
    char *save;
    int argn;
    static const char delim[] = " \t\n" /* 공백, 탭, 개행으로 이루어진 구분자 선언 */
    int pid, status;

    while(1) { /* 무한 반복 */
        printf("[shell] "); /* 프롬프트 출력 */
        gets(buf);
        argn = 0;

        s = strtok_r(buf, delim, &save); /* 문자열에서 delim을 기준으로 단어를 잘라냄 */
        while(s) {
            args[argn++] = s;
            s = strtok_r(NULL, delim, &save);
        }

        args[argn] = (char *)0; /* 인수가 더 없음을 의미하는 문자 추가 */
        if (argn == 0) /* 명령어가 없으면 루프 계속 */
            continue;
        if (!strcmp(args[0], "quit")) /* 프롬프트로 입력받은 단어가 'quit'이면 while 문 벗어남 */
            break;

        if ((pid=fork()) == -1) /* fork 호출에 실패하면 */
            perror("fork failed");
        else if (pid != 0) { /* 부모 프로세스는 자식 프로세스가 종료되기를 기다림 */
            pid = waitpid(pid, &status, 0);
        } else { /* 자식 프로세스는 execvp를 이용하여 arg[0] 실행 */
            execvp(args[0], args);
        }
    }
    exit(0);
}
```

실습 #1

- 명령어 실행 예시

```
user@localhost:~/Lab05
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[user@localhost Lab05]$ ./main
[shell] ls
file hello lab05.c lab05_1.c main
[shell] ls -l
합계 24
-rw-rw-r--. 1 user user 0 5월 12 14:36 file
drwxrwxr-x. 2 user user 6 5월 12 01:40 hello
-rw-rw-r--. 1 user user 1463 5월 11 22:29 lab05.c
-rw-rw-r--. 1 user user 2367 5월 12 01:32 lab05_1.c
-rwxrwxr-x. 1 user user 13096 5월 12 01:38 main
[shell] rm file
[shell] ls -l
합계 24
drwxrwxr-x. 2 user user 6 5월 12 01:40 hello
-rw-rw-r--. 1 user user 1463 5월 11 22:29 lab05.c
-rw-rw-r--. 1 user user 2367 5월 12 01:32 lab05_1.c
-rwxrwxr-x. 1 user user 13096 5월 12 01:38 main
[shell] date
2021. 05. 12. (수) 14:37:03 KST
[shell] echo Hi
Hi
[shell] █
```

실습 #1

- 후면 실행 예시

```
[shell] sleep 5 &  
[shell] █
```

- 명령어 순차 실행 예시

```
[shell] ls  
hello lab05.c lab05_1.c main  
[shell] echo Hi; touch newFile; date  
Hi  
2021. 05. 12. (수) 14: 35: 16 KST  
[shell] ls  
hello lab05.c lab05_1.c main newFile  
[shell] █
```

실습 #1

- 입/출력 리다이렉션 예시

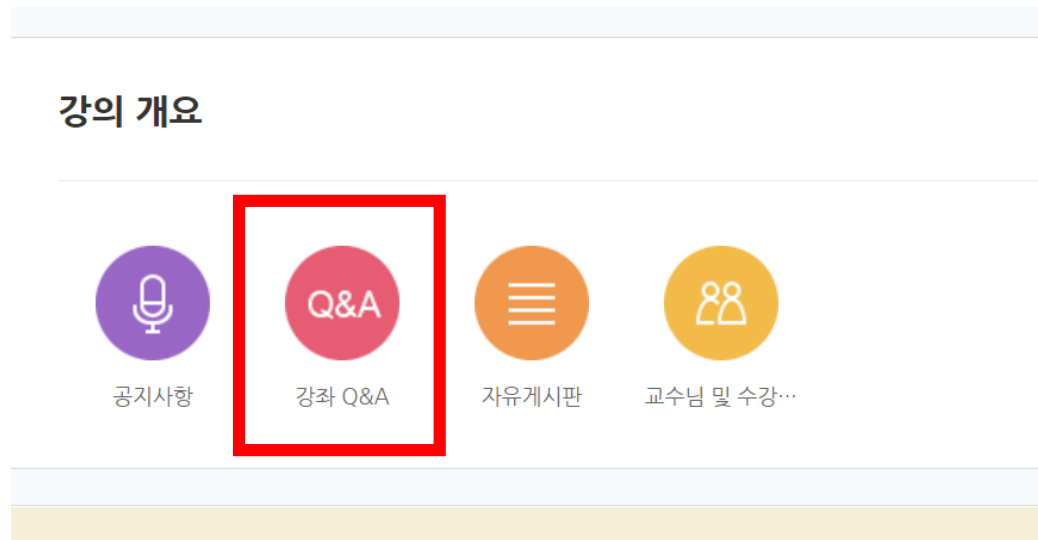
```
[shell] date
2021. 05. 12. (수) 02:00:44 KST
[shell] date > file
[shell] cat file
2021. 05. 12. (수) 02:00:50 KST
[shell] wc < file
  1  6 33
[shell] quit
[user@localhost Lab05] $
```

보고서 구성 및 제출 방법

- 보고서 내용
 - 매주 실습 문제가 적힌 보고서 양식이 제공될 예정입니다.
 - 각 문제별로
 - 캡처 결과
 - 문제 해결 방법: 어떻게 문제를 해결했는지, 프로그램 설계 내용 등 넣어주시기 바랍니다.
- 소스코드(.c, .h 등의 파일들)
 - 컴파일을 하기 위한 모든 소스코드들(사용자 정의 헤더파일이 있는 경우에는 헤더파일도 포함)
- makefile
 - 소스코드들을 컴파일 하기 위한 makefile(소스코드가 하나라도 작성하시기 바랍니다.)
- **위의 3가지를 압축하여** Lab05 에 제출하시기 바랍니다.
- 제출 기한 넘어서 제출한 경우 - 하루 delay 될 때마다 **1점씩 감점**

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 형식으로 보내주세요.