

minios 상에서 Shell command들 구현하기

linux 터미널 상에서 어떤 작업을 하던 가장 기초적으로 필요한 것은 Shell command들입니다.

이번 프로젝트는 OS를 만들어간다는 주제가 강하다고 생각하기에 어느 일정한 부분을 만들어보는 것이 아닌 OS의 가장 기초적인 단계부터 하나씩 쌓아 나가는 과정이 더 중요하다고 생각했습니다.

그렇기에 저희 조는 가장 기초에 속하는 셸 커맨드들을 minios에 구현해보자라는 생각으로 우선적으로 Shell command들을 만들어보기로 하였습니다.

- ls, cd, exit 커맨드 구현

```
minwoo6713@DESKTOP-OD6CBD2: ~/minios$ ./minios
[MiniOS SSU] Hello, World!
>> [SSU OS 8팀] (종료를 원하시면 exit를 입력해주세요.) :ls
kernel lib Makefile scripts minios
drivers newMake include boot README.md
>> [SSU OS 8팀] (종료를 원하시면 exit를 입력해주세요.) :cd
>> [SSU OS 8팀] (종료를 원하시면 exit를 입력해주세요.) :ls
k8s-voting-app ssuOS8_xv6 xv6-public File miniOS
a
>> [SSU OS 8팀] (종료를 원하시면 exit를 입력해주세요.) :exit
minios Shell 커맨드 exit!!
minwoo6713@DESKTOP-OD6CBD2: ~/minios$
```

이후 나머지 mkdir, cp, rm, mv, rmdir, cat 커맨드들을 만들었으나 문제점을 발견하였습니다.

명령을 입력해도 공백란 자체를 하나의 문자열로 인식하여 커맨드를 수행하지 못했습니다. 따라서 사용자가 입력한 명령어를 이해하고, 그 명령어에 따라 적절한 동작을 수행하기 위해서 명령어를 파싱하는 코드를 작성하였습니다.

```
>> [SSU OS 8팀] (종료를 원하시면 exit를 입력해주세요.) :mkdir newproject
>> [SSU OS 8팀] (종료를 원하시면 exit를 입력해주세요.) :ls
kernel lib      Makefile      scripts minios
newproject      drivers newMake  include boot
README.md
```

```
C: parisc X
kernel> Shell-ond> C: parisc
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include "common.h"
5
6 SYMBOL parse(int *waitpid, BOOLEAN makepipe, int *pipefd)
7 {
8     SYMBOL symbol, term;
9     int argc, sourcefd, destfd;
10    int pid, pipefd[2];
11    char *argv[MAXARG+1], sourcefile[MAXNAME];
12    char destfile[MAXNAME];
13    char word[MAXWORD];
14    BOOLEAN append;
15
16    argc = 0;
17    sourcefd = 0;
18    destfd = 1;
19
20    while (TRUE)
21    {
22        switch (symbol = getsymbol(word))
23        {
24            case S_WORD :
25                if (argc == MAXARG)
26                {
27                    fprintf(stderr, "Too many args.\n");
28                    break;
29                }
30                argv[argc] = (char *) malloc(strlen(word)+1);
31                if (argv[argc] == NULL)
32                {
33                    fprintf(stderr, "Out of arg memory.\n");
34                    break;
35                }
36                strcpy(argv[argc], word);
37                argc++;
38            }
39        }
40    }
41    return 0;
42 }
```

- Qemu Shell command 분석

저희가 만든 minios 상의 셸 커맨드와 다르게 어떤 부분이 다른지 체크해보았습니다.

```
선택 minwoo6713@DESKTOP-0D6CBD2: ~/ssuOS8_xv6
Password: 1234
init: starting sh
$ ls
.          1 1 512
..         1 1 512
README    2 2 2286
list.txt   2 3 52
cat        2 4 15548
echo       2 5 14424
forktest   2 6 8876
grep       2 7 18392
init       2 8 15072
kill       2 9 14512
ln         2 10 14408
ls         2 11 16980
mkdir     2 12 14532
rm        2 13 14512
sh        2 14 28572
stressfs   2 15 15444
usertests  2 16 62044
wc        2 17 15968
zombie     2 18 14092
```

- . 현재 디렉토리.

파일 시스템에서 현재 디렉토리를 나타냅니다.

- .. 상위 디렉토리.

파일 시스템에서 상위 디렉토리를 나타냅니다.

xv6 파일 시스템에 있는 예제 파일로, 내용을 열람할 수 있습니다.

- cat

파일 내용을 출력하는 명령어.

cat filename 형식으로 사용되며, 파일의 내용을 표준 출력으로 출력합니다.

- echo

텍스트를 출력하는 명령어.

echo text 형식으로 사용되며, 입력된 텍스트를 표준 출력으로 출력합니다.

- forktest

프로세스 포킹 테스트 프로그램.

fork 시스템 호출을 테스트하기 위해 사용됩니다.

- grep

파일 내에서 문자열을 검색하는 명령어.

grep pattern filename 형식으로 사용되며, 지정된 패턴이 포함된 라인을 출력합니다.

- init

시스템 초기화 프로그램.

시스템 부팅 시 처음으로 실행되는 프로그램입니다.

- kill

프로세스를 종료하는 명령어.

kill pid 형식으로 사용되며, 지정된 프로세스를 종료합니다.

- ln

파일에 대한 링크를 만드는 명령어.

ln target linkname 형식으로 사용되며, 지정된 파일에 대한 하드 링크를 만듭니다.

- ls

디렉토리의 파일 목록을 출력하는 명령어.

ls 형식으로 사용되며, 현재 디렉토리 또는 지정된 디렉토리의 파일 목록을 출력합니다.

- mkdir

새로운 디렉토리를 만드는 명령어.

mkdir dirname 형식으로 사용되며, 지정된 이름의 새 디렉토리를 만듭니다.

- rm

파일을 삭제하는 명령어.

rm filename 형식으로 사용되며, 지정된 파일을 삭제합니다.

- sh

xv6 셸 프로그램.

사용자가 명령어를 입력하고 실행할 수 있는 기본 셸입니다.

- stressfs

파일 시스템 스트레스 테스트 프로그램.

파일 시스템의 성능 및 안정성을 테스트하기 위해 사용됩니다.

- usertests

사용자 테스트 프로그램.

다양한 시스템 호출 및 기능을 테스트하기 위한 프로그램 모음입니다.

- wc

단어 수를 세는 명령어.

wc filename 형식으로 사용되며, 지정된 파일의 단어, 줄, 바이트 수를 출력합니다.

- zombie

좀비 프로세스를 테스트하는 프로그램.

좀비 프로세스가 어떻게 동작하는지 테스트하기 위해 사용됩니다.

minios 대신 xv6 환경에서 앞으로의 프로젝트를 진행하기로 결정하였고,
저희가 만든 minios에서의 셸 커맨드들 대신 xv6에 기본 내장 되어있는 셸 커맨드들을 그대로 이용하기로 결정하였습니다.
기능은 똑같지만 굳이 이미 만들어본 커맨드들을 다시 xv6에서 만들 필요는 없다고 판단하였고, xv6에 내장되어 있는 커맨드들로 충분하다고 판단하였습니다.