

Mobile OS

Homework #1

-simple shell

Junsang Jung

Dankook University

2018 fall

Mobile OS : simple shell

Junsang Jung 32157545

1 Project Introduction : Making a simple shell that can perform the following operations

Entering the executable filename

Finishes execution when it gets 'quit' string from the user

Execution: it has to execute every single executable program in the filesystem, if it has proper privilege

Execution path: to simplify the filename SISH should look into directories in PATH environment variable

During the execution of the user-input program, shell should not be active

Repetition when the given program completes its execution it receives the next input string to run another program

(optional) you can specify different shell prompt using getenv function.

2 Motivation

The SISH mimics the operations of a Linux process, so it is good to understand the operational behavior inside Linux process.

Fork 함수를 통해 child 에서 execute 하는 것에 익숙해 질 수 있다.

3 Concepts used in SISH

Fgets : input string function

We need to write a command. It can accept spaces.

Token : Separate spaces/colon in strings

space를 기준으로 명령어와 파일 이름을 구분하여 token 배열에 입력 받은 순서대로 저장한다.

명령어의 주소들을 colon 기준으로 구분하여 각 주소들을 배열에 저장한다.

Getenv : get environment variable address

명령어의 주소들을 받아와 stat 함수로 받아온 주소에 명령어의 존재를 확인한다.

Stat : get the status of files

Getenv 함수를 통해 받아온 주소에 입력 받은 command 파일이 존재하는 확인한다.

Fork : fork function creates a copy process(child) for the parents process

포크 함수를 통해 하나의 작업이 끝난 후 무한 반복 작업을 하기 위해 child process에 execute 시키고 parent process는 무한loop 한다.

Parent process 는 child process 가 작업을 끝날 때 까지 wait()을 시켜준다.

Execec : run another program by loading the filename.

Child process 를 통해 execute 시키기 위해 execec 함수를 이용하여 입력 받은 명령어를 execute 시킨다.

4 Program Structure

Program run in a loop in the main function

첫번째로 띄어쓰기를 입력받아야 하기 때문에 fgets 함수를 이용하여 입력을 받는다.

입력을 명령어 공백 파일명 순으로 입력을 하기 때문에 배열을 만들어 순서대로 token

화 시켜서 저장한다.

명령어들의 주소를 getenv 함수를 통해 가져온다.

명령어들의 주소를 가져오면 한줄에 colon을 기준으로 출력되는 것을 colon 기준으로 token 화 하여 주소값을 구분지어준다.

strcat함수를 이용하여 받아온 환경변수 주소뒤에 /입력받은명령어 를 붙여 준다.

Stat 함수를 이용하여 입력한 명령어가 환경변수+/입력받은 명령어 주소로 찾아가 존재 하는지 확인한다.

Fork 함수를 이용하여 parent process 와 child process를 만들어 준다.

execev 를 이용하여 Child process 에 입력받은 command 를 실행시켜준다.

Parent process 는 child process 가 끝난뒤 다시 명령어를 입력받기 위해 wait()를 시켜 child process 가 입력받은 명령어를 수행할때까지 기다린다.

If 문을 이용하여 입력받은 명령어가 대문자인지 소문자인지 구분해준다.

입력받은 명령어가 소문자라면 위의 structure 를 따라 process 가 실행되고

입력받은 명령어가 대문자라면 바로 getenv 함수에 입력받은 환경변수를 넘겨주어 printf 실행시켜 준다.

Quit 를 입력하여 SISH 를 종료 할 수 있게 해준다.

5 Problem and solutions

Getenv 함수로 path 를 가져온뒤 뒤에 / 입력받은 명령어 를 붙여 stat 시키는 과정에서 path가 1번부터 11번까지 확인을 하지 못하는 경우가 발생했습니다.

그래서 copy시켜 copy를 stat 시켜 명령어의 존재하는 주소를 찾는 작업을 실행했습니다.

6 Build environment

Compilation : Linux Assam, with GCC

Sish_32157545.out

./sish_32157545

7 Screen capture

```
jjs15@assam:~/2018_os_hw1$ ./sish_32157545
jjs15@SiSH:ls
README.md  fork.c      hw1.pdf     shell2.c    sish_32157545
a.out      getenv.c    shell.c      shell2.out  stat.c
exam.c     getenv.out  shell.out    shell3.c    stat.out
jjs15@SiSH:mkdir abc.txt
jjs15@SiSH:ls
README.md  exam.c      getenv.out  shell.out    shell3.c    stat.out
a.out      fork.c      hw1.pdf     shell2.c     sish_32157545
abc.txt    getenv.c    shell.c      shell2.out   stat.c
jjs15@SiSH:quit
jjs15@assam:~/2018_os_hw1$
```

./sish_32157545 로 실행

Ls 명령어 입력시 파일 리스트 표시

Mkdir 명령어로 txt 파일 생성

Quit 입력으로 sish 종료

8 Personal feelings

Sish를 통해 fork 함수를 이용하여 하나의 process 가 종료되어도 실행중인 파일이 종료되지 않게 할 수 있는 process 를 알수 있게 되었고, getenv 함수를 통해 명령어의 주소를 token 화 시켜 stat 시켜 명령어를 실행시키는 process를 알수 있게 되었다.

이를 통해 Linux 의 command process 가 어떻게 돌아가는지에 대한 이해를 할수 있었고 또한 개인적으로 Linux 환경에 익숙해지고 c언어의 함수들을 배울수 있었다.