

Report

System Programing

Assignment #1

수업	시스템프로그래밍실습
수업시간	월 1, 2
O S	Ubuntu 12.04 64bit
과제번호	Assignment #1
제출날짜	2014-04-18
전공	컴퓨터공학과
학번	2009720151
이름	이재해

<Program Overview>

-프로그램소개

- *linux의 terminal에서 사용되는 명령어들을 구현하는 프로그램입니다.
- *사용 가능한 명령어는 ls, dir, cd, mkdir, delete, rmdir, rename이 있습니다.
- *ls는 a옵션이나 l옵션의 사용이 가능합니다.
옵션을 입력할 때에 중복되거나 순서가 뒤바뀌어도 상관없습니다.
- *다른 명령어에는 옵션이 존재하지 않습니다.
- *이 프로그램의 client에서 명령어를 구분하고 에러를 찾아내고 server에서 사용가능하도록 정리합니다.
- *이 프로그램의 server에서는 정리된 명령어를 이용해서 결과물을 출력만합니다.
- *client에서 명령어를 정리해서 정리된 명령어를 server로 전송하여 server에서 결과물을 출력합니다.

-문제분석

이번 문제는 c언어로 linux의 명령어를 구현하는 프로그램을 만드는 것입니다. 이 명령어를 구현하는 방식은 client에서 명령어를 입력받아 입력받은 명령어를 정리하고 server로 전송하여 정리된 명령어에 따라서 server에서 그 명령어가 원하는 행동을 하는 취하는 것입니다.

여기서 client와 server가 정리된 명령어의 전송 방식은 read함수와 write함수를 이용하여 전송됩니다. 하지만 해당 프로그램에서는 그 client에서 write함수를 이용하여 화면에 정리된 명령어가 출력되게 되고, server에서는 write를 이용하여 데이터를 입력받아야 합니다. 이 때 리눅스 명령어인 tee를 이용하여 화면의 출력 값을 입력 값으로 입력되게 합니다.

이 프로그램에서 명령어가 잘못됐을 경우 에러처리를 해줘야합니다.

client에서는 server에서 사용가능하도록 명령어를 정리하는 방식은 입력된 데이터를 argc와 argv를 이용해서 정리해줍니다. 정리하는 도중에 명령어가 잘못됐으면 에러처리를 해줘야 합니다. 만약 명령어가 사용이 불가능한 에러라면 server에서 명령어를 수행하지 못하게 에러만 출력하게 명령어를 정리해줍니다.

이번 프로그램을 만들 때 사용이 금지된 함수들이 있습니다.

먼저 printf() in <stdio.h>, fprintf() in <stdio.h>, fputs() in <stdio.h> 같이 system call인 write()로 대체가 가능하다면 write()함수를 이용해야합니다. 마찬가지로 scanf() in <stdio.h>, fscanf() in <stdio.h>, fscanf() in <stdio.h> 같이 system call인 read()로 대체가 가능하다면 read()함수를 이용해야 합니다. 특히나 ls의 -l 옵션을 출력할 때에 sprintf()함수를 이용하면 간단히 사용이 가능하지만 이 함수 또한 write()함수로 대체가 가능하기 때문에 이번 프로그램에서는 사용할 수 없습니다.

마찬가지로 ls의 옵션을 구별하기 위해서는 원래 getopt()를 이용해서 옵션을 나누면 간단하게 처리가 가능하지만 이번 프로그램에서는 getopt를 이용이 불가능해서 일일이 비교해가면서 옵션을 정리해 주어야 합니다.

또한 ls나 dir에서 wildcard(*)를 사용하기 때문에 wildcard를 간편하게 처리해주는 fnmatch() in <fnmatch.h>. glob() in <glob.h>같은 함수들의 사용을 금지하고 wildcard를

따로 처리해주어야 합니다.

이 외에도 remove() in <stdio.h> 같이 rmdir(), unlink()로 대체할 system call이 존재하는 경우에는 rmdir(), unlink()를 이용해서 제작해야 합니다.

-실행 결과 캡처 화면

```
lee@ubuntu: ~/SS
.srv: command not found
lee@ubuntu:~/SS$ ./cli cd 351 | tee cli.out | .srv
.srv: command not found
lee@ubuntu:~/SS$ ./cli dir | tee cli.out | .srv
drwxrwxrwx    7    lee    lee    4096    APR 18 7:46    ./
drwxr-xr-x   27    lee    lee    4096    APR 18 7:41    ../
drwxrwxr-x    6    lee    lee    4096    APR 18 7:46    1/
drwxrwxr-x    2    lee    lee    4096    APR 18 6:50    14/
drwxrwxr-x    2    lee    lee    4096    APR 18 6:51    16/
drwxrwxr-x    2    lee    lee    4096    APR 17 21:55    6/
drwxrwxr-x    2    lee    lee    4096    APR 17 17:54    8/
-rw-rw-r--    1    lee    lee    22     APR 17 17:17    c.out
-rwxrwxr-x    1    lee    lee    29789   APR 18 7:46    cli
-rw-rw-r--    1    lee    lee    16158   APR 18 7:46    cli.c
-rw-rw-r--    1    lee    lee    16144   APR 18 7:42    cli.c~
-rw-rw-r--    1    lee    lee    4       APR 18 7:53    cli.out
-rw-rw-r--    1    lee    lee    73     APR 17 17:17    Makefile
-rw-rw-r--    1    lee    lee    29     APR 17 17:17    Makefile1
-rw-rw-r--    1    lee    lee    0       APR 17 17:16    Makefile~
-rwxrwxr-x    1    lee    lee    30383   APR 18 7:39    srv
-rw-rw-r--    1    lee    lee    15255   APR 18 7:39    srv.c
-rw-rw-r--    1    lee    lee    15269   APR 18 7:38    srv.c~
lee@ubuntu:~/SS$ ./cli cd | tee cli.out | .srv
Error: another argument is required
lee@ubuntu:~/SS$
lee@ubuntu:~/SS$
lee@ubuntu:~/SS$ ./cli cd asdf | tee cli.out | .srv
bash: cd: asdf: No such file or directory
lee@ubuntu:~/SS$ ./cli cd 14 | tee cli.out | .srv
lee@ubuntu:~/SS$ ./cli cd cli | tee cli.out | .srv
bash: cd: cli: Not a directory
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | .srv
1/          8/          cli.c          Makefile          srv
14/         c.out       cli.c~         Makefile1         srv.
c
16/         cli          cli.out        Makefile~         srv.
c~
6/
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | .srv
1/          8/          cli.c          Makefile          srv
14/         c.out       cli.c~         Makefile1         srv.c
16/         cli          cli.out        Makefile~         srv.c~
6/
lee@ubuntu:~/SS$
```

```
lee@ubuntu: ~/SS
lee@ubuntu:~/SS$ ./cli ls '*.c'
c.
NLST cli.c srv.c
lee@ubuntu:~/SS$ make
gcc cli.c -o cli
lee@ubuntu:~/SS$ ./cli ls '*.c' | tee cli.out | ./srv
cli.c srv.c
lee@ubuntu:~/SS$ ./cli ls 'cl*' | tee cli.out | ./srv
cli cli.c cli.c~ cli.out
lee@ubuntu:~/SS$ ./cli pwd | tee cli.out | ./srv
/home/lee/SS
lee@ubuntu:~/SS$ ./cli cd | tee cli.out | ./srv
Error: another argument is required
lee@ubuntu:~/SS$ ./cli cd .. | tee cli.out | ./srv
lee@ubuntu:~/SS$
lee@ubuntu:~/SS$ ./cli cd /home/lee | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls -a | tee cli.out | ./srv
./ 16/ cli Makefile srv
./ 6/ cli.c Makefile1 srv.c
1/ 8/ cli.c~ Makefile~ srv.c~
14/ c.out cli.out
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/ 8/ cli.c Makefile srv
14/ c.out cli.c~ Makefile1 srv.c
16/ cli cli.out Makefile~ srv.c~
6/
lee@ubuntu:~/SS$ ./cli mkdir mk | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/ 8/ cli.c~ Makefile1 srv
14/ c.out cli.out Makefile~ srv.c
16/ cli Makefile mk/ srv.c~
6/ cli.c
lee@ubuntu:~/SS$ ./cli rmdir mk | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/ 8/ cli.c Makefile srv
14/ c.out cli.c~ Makefile1 srv.c
16/ cli cli.out Makefile~ srv.c~
6/
lee@ubuntu:~/SS$ ./cli delete | tee cli.out | ./srv
delete: missing operand
delete: usage: delete [Arguments]
lee@ubuntu:~/SS$ ./cli delete 8 | tee cli.out | ./srv
delete: cannot remove '8': Is a directory
lee@ubuntu:~/SS$
```

```
lee@ubuntu: ~/SS
gcc cli.c -o cli
lee@ubuntu:~/SS$ ./cli mkdir a b c d
MKD      a      b      c      d
lee@ubuntu:~/SS$ ./cli mkdir a b c d | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/      6/      c/      cli.out Makefile~
14/     a/      cli      d/      srv
16/     b/      cli.c    Makefile      srv.c
1a/     c.out   cli.c~   Makefile1     srv.c~
lee@ubuntu:~/SS$ ./cli rmdir a b c d | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/      6/      c/      cli.out Makefile~
14/     a/      cli      d/      srv
16/     b/      cli.c    Makefile      srv.c
1a/     c.out   cli.c~   Makefile1     srv.c~
lee@ubuntu:~/SS$ ./cli rmdir a b c d | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/      6/      cli.c    Makefile      srv
14/      c.out   cli.c~   Makefile1     srv.c
16/      cli      cli.out   Makefile~     srv.c~
1a/
lee@ubuntu:~/SS$ ./cli mkdir a b c d | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/      6/      c/      cli.out Makefile~
14/     a/      cli      d/      srv
16/     b/      cli.c    Makefile      srv.c
1a/     c.out   cli.c~   Makefile1     srv.c~
lee@ubuntu:~/SS$ ./cli rename a a_change | tee cli.out | ./srv
lee@ubuntu:~/SS$ ./cli ls | tee cli.out | ./srv
1/      6/      c/      cli.out Makefile~
14/     a_change/ cli      d/      srv
16/     b/      cli.c    Makefile      srv.c
1a/     c.out   cli.c~   Makefile1     srv.c~
lee@ubuntu:~/SS$ ./cli rename a -b a_change | tee cli.out | ./srv
rename: cannot move '-b' to a subdirectory of itself, '-b/-b'
lee@ubuntu:~/SS$ ./cli ls -e | tee cli.out | ./srv
ls: invalid option -- 'e'
ls: Available options: a l
lee@ubuntu:~/SS$ ./cli ls -l cli | tee cli.out | ./srv
-rwxrwxr-x  1      lee      lee      29789  APR 18 8:12  cli
lee@ubuntu:~/SS$ ./cli ls -l cli srv | tee cli.out | ./srv
-rwxrwxr-x  1      lee      lee      29789  APR 18 8:12  cli
-rwxrwxr-x  1      lee      lee      30383  APR 18 7:39  srv
lee@ubuntu:~/SS$
```

<Algorithm>

-pseudo code

이번 프로그램의 소스의 함수를 세분화 하지 않아서 중요하다고 생각하는 일부분의 소스만을 이용해서 pseudo code를 만들었습니다.

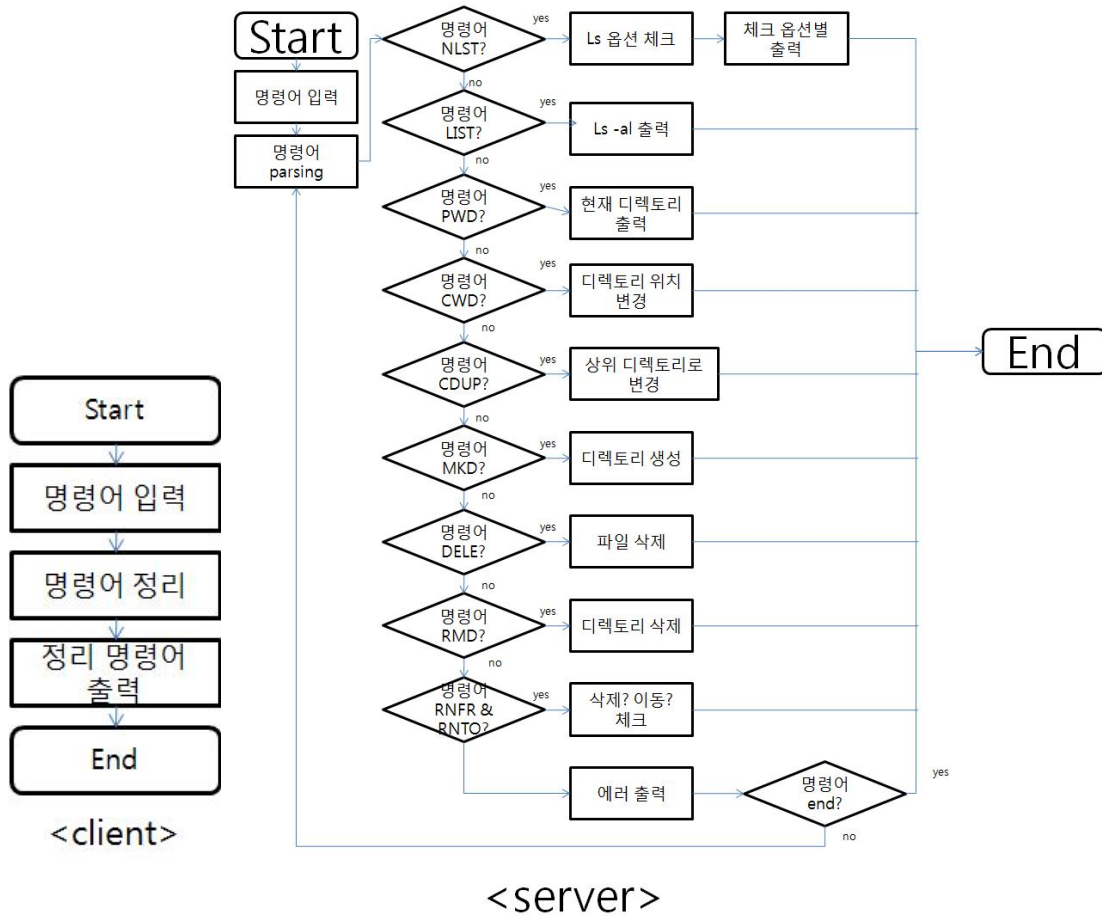
```
char getStr[128];
char data[1024][128];
int i=0, count=0;
struct dirent *directory entry;
DIR *directory pointer;
struct stat data's stat;

getStr get current working directory;
directory pointer=current working directory pointer;

while (directory entry = read directory(directory pointer))
{
    if(directory entry->name first character is "." Not)
    {
        copyt directory entry->name to data[i]
        read data[i]'s stat to data's stat
        if(data is directory)
            data[i] add /;
        count plus 1; i plus 1;
    }
}
clear directory pointer;
```

위 슈도코드는 데이터를 현재 디렉토리의 정보를 읽어와서 현재 디렉토리에 있는 파일들을 읽어 들이는 가장 기본적인 코드입니다. 이 코드는 ls를 만드는데 기본적으로 필요한 코드입니다. 이 코드에서 while안에 directory entry에 저장돼 있는 이름의 첫 번째 character가 “.”(숨김 파일 구분)이면 숨김 파일로 데이터 array에 저장되지 않아 나중에 출력 되지 않습니다. 여기서 숨김 파일을 구분해주는 “.”인지 아닌지를 확인해 주는 if문만 지워주면 출력할 때에 숨김 파일까지 출력시켜줄 수 있습니다.

-flow chart



<Conclusion and discussion>

이번 Assignment는 linux의 terminal에서 사용되는 명령어들을 구현하는 프로그램을 만드는 것이었습니다. 리눅스 환경을 이번 학기에 와서 처음 경험해 봐서 조금 힘든 부분들이 있었습니다. 또한 이러한 프로그램의 입력 방식을 처음 사용해봐서 처음에 프로그램을 설계할 때와 프로그램 만든 다음의 생각이 많이 바뀌게 되었습니다.

처음에 설계할 당시 강의자료에서 client에서 에러를 출력한 모습을 보고 모든 에러처리를 client에서 하고 server에서는 명령어를 그저 수행만 시키려고 계획했었는데 만드는 과정에서 server에서 system call을 사용하는 과정에서 에러가 날 때 에러처리를 했다면 제가 만든 것보다는 쉽게 처리하는게 가능했었다는 것을 알게 되었지만 다시 제작하기에는 시간이 부족하게 되어 그냥 client에서 가능한 만큼 에러처리를 해 두었습니다.

또한 이미 프로그램을 만들고 난 뒤에 제안서 안에서 Data Structure Diagram을 보고 Data Structure를 이용해서 linked list를 만들어서 제작하면 되는 것을 알게 됐지만 이 부분 또한 이미 다시 제작하기에는 시간이 부족하게 되어서 Data Structure를 사용하지 않고 제작하게 되었습니다.

이번 과제는 너무 초반의 프로그램을 제작하는데 필요한 기본적인 공부 부족과 제안서를 꼼꼼히 살펴보지 못해 프로그램의 부족한 부분을 계속 추가해서 만들다보니 client와 server의 소스 두 가지 모두 주석이 없이 이미 800줄이 넘어있던 상황이 되었습니다. 한번 만들어보니 어떠한 방식으로 짜면 심플하게 만들 수 있을지를 알 수 있었습니다.