

System Programming

(Assignment2)

과 목	시스템프로그래밍실습
담당교수	이기훈 교수님
학 과	컴퓨터공학과
학 번	2010720149
성 명	이동현
날 짜	2016. 04. 01 (목)

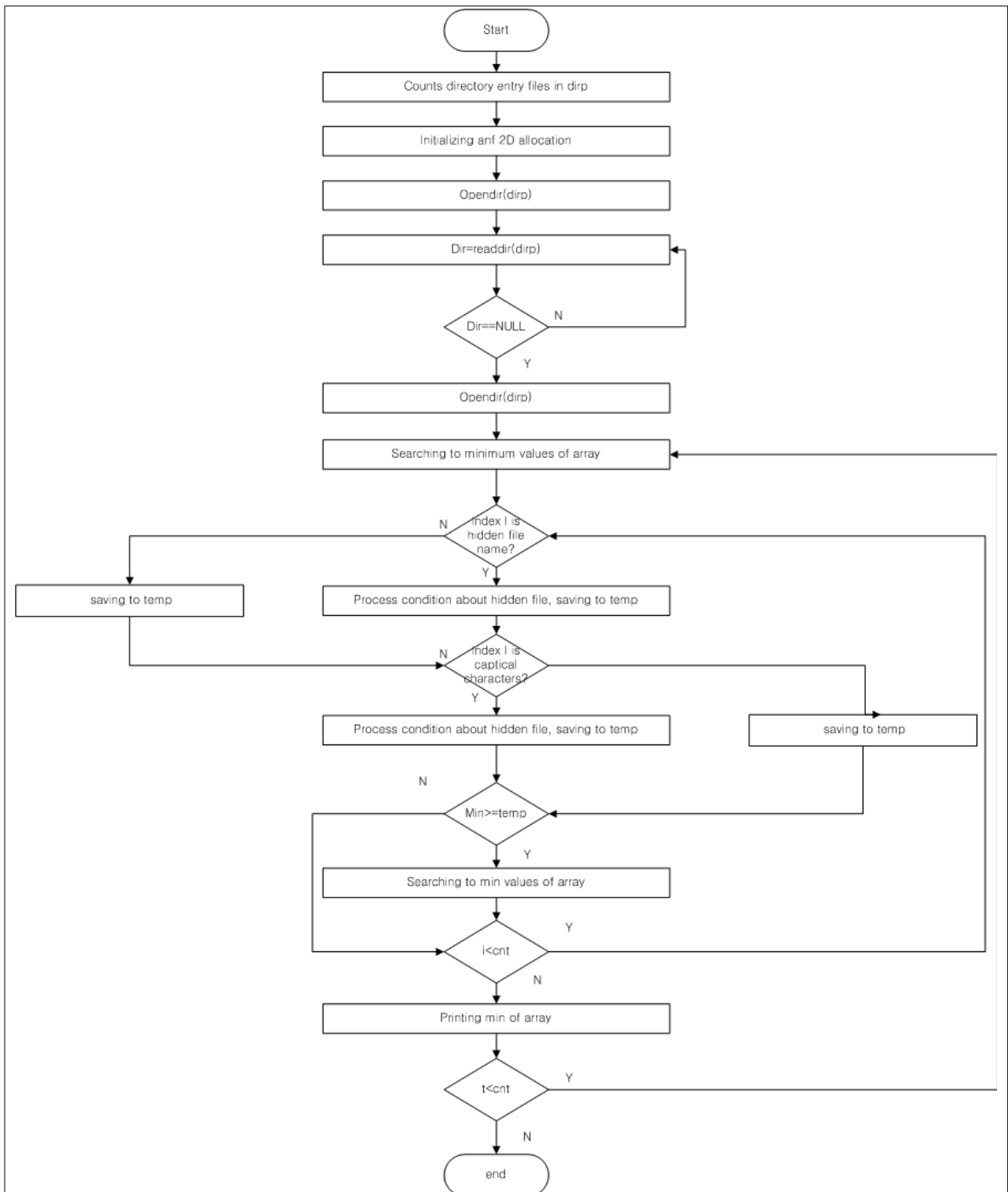


A. Introduction

♣ Simple ls를 구현하기 ♣

Shell에서 현재 Working directory list를 출력하는 명령어인 ls를 직접 코드로 구현하여 명령어의 기능과 그에 맞는 c코드를 직접 구현하여 명령어에 대해 이해하는데 목적이 있다.

B. Flowchart



C. Pseudo code

```

int main
{
    set file of dir cnt
    allocate save[]
    dirp=opendir(".");
    for(dir=readdir(dirp) ; dir ; dir=readdir(dirp))
        save[i], save dir ->d_name;
        i++
    closedir(dirp);
    while(t<cnt)
        for(z=0 ; z<cnt ; z++)
            searching to min
            for(i=0 ; i<cnt ; i++)
                if(save[i]!=NULL)
                    if(hidden is)
                        for(j=0 ; save[i][j]!=0 ; j++)
                            temp[j]<=save[i][j+1];
                        for(f=0 ; temp[f]!=0 ; f++)
                            temp[f] change to small letter
                    else
                        temp <- save[i]
                        same set, change to small letter
                if(min>=temp)
                    min<-temp
                    k=i
            printf("%s", save[i]);
            save[k]<- NULL
            t++
            initialzing temp

    free(save)
}

```

D. Reference

- 강의자료 ' 2016-1_SPLab_04_basic_v3 '

E. Conclusion

♣ 조건 ♣

- 파일 이름만 출력
- 히든파일 포함(.)
- 한 줄에 하나의 파일 출력
- 파일 이름을 정렬하여 출력

♣ 결과 화면 1 ♣

```
ldh@ubuntu:~/Sys/assign2$ ls -al
total 64
drwxrwxr-x 3 ldh ldh 4096 Mar 31 06:38 .
drwxrwxr-x 6 ldh ldh 4096 Mar 30 20:52 ..
-rw-rw-r-- 1 ldh ldh 0 Mar 30 12:38 _dd
-rw-rw-r-- 1 ldh ldh 1168 Mar 30 04:39 ex.c
drwxrwxr-x 2 ldh ldh 4096 Mar 31 06:38 Folder
-rw----- 1 ldh ldh 12288 Mar 27 06:13 .gedit.swp
-rw-rw-r-- 1 ldh ldh 0 Mar 30 12:38 ..kk
-rw-rw-r-- 1 ldh ldh 1755 Mar 30 12:34 ls.c
-rw-rw-r-- 1 ldh ldh 2136 Mar 30 12:27 ls.c~
-rw-r--r-- 1 ldh ldh 12288 Mar 24 07:53 .ls.c.swp
-rw-rw-r-- 1 ldh ldh 50 Mar 30 12:46 Makefile
-rw-rw-r-- 1 ldh ldh 44 Mar 30 04:36 Makefile~
-rwxrwxr-x 1 ldh ldh 7472 Mar 30 12:46 spls
ldh@ubuntu:~/Sys/assign2$ ./spls
.
..
..kk
_dd
ex.c
Folder
.gedit.swp
ls.c
.ls.c.swp
ls.c~
Makefile
Makefile~
spls
ldh@ubuntu:~/Sys/assign2$ █
```

Shell 명령어 `ls -al`을 통해 현재 디렉토리의 list를 출력하였다. 내가 구현한 `spls` 실행파일을 실행한 결과, ...가 먼저 출력되고 대소문자 상관없이 정렬이 되었음을 확인 할 수 있다. 단, 히든파일은 맨 앞자리 .을 무시하고 정렬하도록 조건에 맞게 구현을 하였다.

♣ 결과 화면 2 ♣

```
ldh@ubuntu:~$ ./spls
.
..
.bash_history
.bash_logout
.bashrc
.cache
.config
.dbus
Desktop
.dmrc
Documents
Downloads
Embe
examples.desktop
.fontconfig
.gconf
.gksu.lock
.gnome2
.gstreamer-0.10
.gtk-bookmarks
.gvfs
.ICEauthority
.local
ls.c
Makefile
Makefile~
.mission-control
.mozilla
Music
Pictures
.profile
.Public
.pulse
.pulse-cookie
qt-everywhere-opensource-src-5.4.2
qt-everywhere-opensource-src-5.4.2.tar.gz
spls
Sys
Templates
.thumbnails
Videos
.viminfo
.Xauthority
.xsession-errors
.xsession-errors.old
ldh@ubuntu:~$ █
```

Makefile

```
ldh@ubuntu: ~/Sys/ass
spls : ls.c
      gcc ls.c -o spls
clean:
      rm -rf spls
~
```

- 많은 데이터를 가진 /home/ldh 디렉토리에서 결과 화면.

♣ 고찰 ♣

이번 시스템프로그래밍 실습 2차 과제는 shell 명령어인 ls를 c언어로 구현하였다. ls는 현재 디렉토리의 list를 출력하는 명령어이다. 파일입출력과 같이 DIR 구조체에서 현재 디렉토리를 받고, dirp로 현재 디렉토리를 읽어와 처리하는 방식이다. file stream에 관하여 구현을 해보았기 때문에, directory stream을 사용하는데 큰 어려움이 없었다. 그리고 정렬은 임의의 배열에 저장하여 min값을 키로 두어 min값을 구현하는 방법을 이용하였다. min값을 찾기전에 hidden파일과 대문자를 확인하여, 임의의 배열에 다시 저장하고 그 배열을 통해 min값을 찾는 방식의 알고리즘을 사용하였다.

그러나 이번 과제를 하면서 리눅스를 통한 c구현이기 때문에, 디버깅과정이 쉽지 않아 고생을 한 것 같다. 아직 gdb가 익숙지 않고 printf를 통해 자체 디버깅을 하였기 때문에, 주소값 접근 에러를 찾는데 난항을 겪어 시간 소비가 꽤 있었다. 긴 코드를 디버깅을 할 때, gdb 디버깅 방법에 익숙할 필요가 있다고 생각이 들었다. 그래도 segment fault가 발생하더라도 리눅스는 오류가 아닌 에러 발생 전까지 출력 화면을 제공을 해주기 때문에 간단한 알고리즘에 대한 에러나 현재 구성도를 확인하기 위한 방법은 printf 방식도 괜찮다고 생각이 들었다.

그 이외에는 많이 접했던 코드 방식이라 큰 어려움이 없었고 앞으로 ls의 옵션과 파일의 정보 등을 구현하기 위해 새로운 구조체를 많이 사용할 예정인데, ls 각각의 옵션을 이해하여 앞으로 구현에 있어 도움이 되도록 준비를 해야겠다.