KWANGWOON UNIVERSITY

2019년 1학기 **시스템프로그래밍실습** 5주차

Advanced ls

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Contents

- ls
- Data Types
- System Call for *Getting File Information*
- Function for Getting User Information
- Function for Getting Group Information
- Function for Parsing Time Information
- System Call for Getting Current Working Directory
- Function for Parsing Program Execution Options

- Lab.

- Assignment 2-2

ls

- **Data types**
  - DIR, struct dirent, **struct passwd, struct stat, struct tm**

- **System Calls & Functions**
  - **opendir(), readdir(), closedir()**
  - **stat()**
  - **getgrgid(), getpwuid()**
  - **localtime()**
  - **getcwd()**
  - **getopt()**
  - **fnmatch()**

Data types

- **Header** : <pwd.h>
- **Data type** : struct **passwd**
- **Members**
  - char *pw_name; // user name
  - char *pw_passwd; // user password
  - uid_t pw_uid; // user ID
  - gid_t pw_gid; // group ID
  - char *pw_gecos; // user information
  - char *pw_dir; // home directory
  - char *pw_shell; // shell program

- **Passwd file** : /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
```

Data types (cont'd)

- **Header** : <grp.h>
- **Data type** : struct **group**
- **Members**
  - char *gr_name; // group name
  - char *gr_passwd; // group password
  - gid_t gr_gid; // group ID
  - char **gr_mem; // group members

- **Group file** : /etc/group

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:sslab
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
```

Data types (cont'd)

- **Header** : <sys/stat.h>
- **Data type** : struct **stat**
- **Members**
  - …
  - ino_t st_ino; // inode number
  - mode_t st_mode; // protection
  - nlink_t st_nlink; // number of hard links
  - uid_t st_uid; // user ID of owner
  - gid_t st_gid; // group ID of owner
  - …
  - off_t st_size; // total size, in bytes
  - blksize_t st_blksize; // block size for file system I/O
  - blkcnt_t st_blocks; // number of 512B blocks allocated
  - time_t st_atime; // time of last access
  - time_t st_mtime; // time of last modification
  - time_t st_ctime; // time of last status change

Data types (cont'd)

- **Header** : <time.h>
- **Data type** : struct **tm**
- **Members**
  - int tm_sec; // # of seconds after the minute [0-60]
  - int tm_min; // # of seconds after the hour [0-59]
  - int tm_hour; // # of hours past midnight [0-23]
  - int tm_mday; // The day of the month [1-31]
  - int tm_mon; // # of months since January [0-11]
  - int tm_year; // # of years since 1900
  - int tm_wday; // # of days since Sunday [0-6]
  - int ym_yday; // # of days since January 1 [0-365]

System Call for Getting File Information

- **`stat()`**

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

int stat(const char *path, struct stat *buf);
```

  - Description
    - This function returns information about a file.
    - It stats the file pointed to by `path` and fills in `buf`.
  - Return value
    - On success, 0
    - On error, -1

Function for Getting User Information

- **`getpwuid()`**

```
#include <sys/types.h>
#include <pwd.h>

struct passwd *getpwuid(uid_t uid);
```

  - Description
    - It returns a pointer to a structure containing the broken-out fields of the record in the password database that matches the user ID `uid`.

  - Return value
    - On success, a pointer to a `passwd` structure
    - On error, NULL

Function for Getting Group Information

- **`getgrgid()`**

```
#include <sys/types.h>
#include <grp.h>

struct group *getgrgid(gid_t gid);
```

  - Description
    - It returns a pointer to a structure containing the broken-out fields of the record in the group database that matches the group ID `gid`.

  - Return value
    - On success, a pointer to a `group` structure
    - On error, NULL

Function for Parsing Time Information

- **`localtime()`**

```
#include <time.h>

struct tm *localtime(const time_t *timep);
```

  - Description
    - It converts `time_t` to `struct tm` with current time zone and daylight.
  - Return value
    - On success, a pointer to a `tm` structure
    - On error, NULL

System Call for Getting Current Working Directory

- **getcwd()**

```
#include <unistd.h>

char *getcwd(char *buf, size_t size);
```

  - Description
    - It copies an absolute pathname of the current working directory to the array pointed to by `buf`, which is of length `size`.
  - Return value
    - On success
      - A pointer to a string containing the pathname of the current working directory
    - On error
      - NULL

Function for Parsing Program Execution Options

- **getopt()**

```
#include <unistd.h>

int getopt(int argc, char * const argv[], const char
*optstring);
extern char *optarg;
extern int optind, opterr, optopt;
```

  - Description
    - It parses the command-line arguments.
      - Option is started with "-".
    - It changes the contents of `argv` as it scans,
      - Eventually all the **nonoptions are at the end**.
      - In this case, after `getopt` returns -1,
        - **The `optind` is the index in argv of the first argv-element that is not an option.**
    - `const char *optstring`
      - It is a string containing the option characters to use.
      - If such a character is followed by
        - A colon(:),the option requires an argument.

Function for Parsing Program Execution Options (cont'd)

- **getopt()**

```
#include <unistd.h>

int getopt(int argc, char * const argv[], const char
*optstring);
extern char *optarg;
extern int optind, opterr, optopt;
```

  - Return value
    - On success, option character
    - Option is not in `optstring`, '?'
    - All command-line options have been parsed, -1

Function for Parsing Program Execution Options (cont'd)

- **getopt()**

```
#include <unistd.h>

int getopt(int argc, char * const argv[], const char
*optstring);
extern char *optarg;
extern int optind, opterr, optopt;
```

- extern char *optarg;
  - Option's argument value or 0 when the option has no argument
- extern int **opterr**;
  - If `opterr` is nonzero, `getopt` prints error messages when (an) option is not right.
  - `Its default value is 1.`
- extern int **optind**;
  - It is the index of the next element to be processed in `argv`.
- extern int **optopt**;
  - option character passed back to user (i.e. it has unknown(failed) option character.)

Example

- **Source code**

```c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv)
{
        int aflag = 0, bflag = 0;
        char *cvalue = NULL;
        int c = 0;

        while((c = getopt(argc, argv, "abdc:")) != -1)
        {

                //============print the getopt variable=============//
                printf("optarg = %s\toptind = %d\topterr = %d\toptopt = %c\n",\
                        optarg, optind, opterr, optopt);
                //==================================================//

                switch(c)
                {
                case 'a':                       // just flag on
                        aflag++;
                        break;
                case 'b':
                        bflag++;                // just flag on
                        break;
                case 'c':
                        cvalue = optarg;        // get argument
                        break;
                case 'd':
                        opterr = 0;             // opterr off
                        break;
                case '?':                       // print the error message
                        printf("Unknown option character\n");
                        break;
                }
        }
        //============print flags and c's argument============//
        printf("\naflag = %d\tbflag = %d\tcvalue = %s\n", aflag, bflag, cvalue);
        return 0;
}
```

Example (cont'd)

- **Execution**

```
aa2013722077@ubuntu:~/practice/2_ls/test_code$ ./test -a -b -chello -hello
optarg = (null) optind = 2      opterr = 1      optopt =
optarg = (null) optind = 3      opterr = 1      optopt =
optarg = hello  optind = 4      opterr = 1      optopt =
./test: invalid option -- 'h'
optarg = (null) optind = 4      opterr = 1      optopt = h
Unknown option character
./test: invalid option -- 'e'
optarg = (null) optind = 4      opterr = 1      optopt = e
Unknown option character
./test: invalid option -- 'l'
optarg = (null) optind = 4      opterr = 1      optopt = l
Unknown option character
./test: invalid option -- 'l'
optarg = (null) optind = 4      opterr = 1      optopt = l
Unknown option character
./test: invalid option -- 'o'
optarg = (null) optind = 5      opterr = 1      optopt = o
Unknown option character

aflag = 1       bflag = 1       cvalue = hello
```

Example

- **Execution** (cont'd)

```
aa2013722077@ubuntu:~/practice/2_ls/test_code$ ./test -d -a -b -chello -hello
optarg = (null) optind = 2      opterr = 1      optopt =
optarg = (null) optind = 3      opterr = 0      optopt =
optarg = (null) optind = 4      opterr = 0      optopt =
optarg = hello  optind = 5      opterr = 0      optopt =
optarg = (null) optind = 5      opterr = 0      optopt = h
Unknown option character
optarg = (null) optind = 5      opterr = 0      optopt = e
Unknown option character
optarg = (null) optind = 5      opterr = 0      optopt = l
Unknown option character
optarg = (null) optind = 5      opterr = 0      optopt = l
Unknown option character
optarg = (null) optind = 6      opterr = 0      optopt = o
Unknown option character

aflag = 1       bflag = 1       cvalue = hello
```

```
aa2013722077@ubuntu:~/practice/2_ls/test_code$ ./test -abchello
optarg = (null) optind = 1      opterr = 1      optopt =
optarg = (null) optind = 1      opterr = 1      optopt =
optarg = hello  optind = 2      opterr = 1      optopt =

aflag = 1       bflag = 1       cvalue = hello
```

Lab.

- **Source code**
  - Fill the blank!

```c
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv)
{
        int aflag = 0, bflag = 0;
        char *cvalue = NULL;
        int index = 0, c = 0;
        int i = 0;
        opterr = 0;
        while((c = getopt(argc, argv, "abc:"))  != -1)
        {
                switch(c)
                {
                }
        }
        printf("aflag = %d, bflag = %d, cvalue = %s\n", aflag, bflag, cvalue);

        return 0;
}
```

Lab. (cont'd)

- **Execution**

```
aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt
aflag = 0, bflag = 0, cvalue = (null)

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -a -b
aflag = 1, bflag = 1, cvalue = (null)

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -ab
aflag = 1, bflag = 1, cvalue = (null)

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -c foo
aflag = 0, bflag = 0, cvalue = foo

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -cfoo
aflag = 0, bflag = 0, cvalue = foo

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt arg1
aflag = 0, bflag = 0, cvalue = (null)
Non-option argument arg1
```

Lab.

- **Execution** (cont'd)

```
aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -a arg1
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument arg1

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -c foo arg1
aflag = 0, bflag = 0, cvalue = foo
Non-option argument arg1

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -a -- -b
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument -b

aa2013722077@ubuntu:~/practice/2_ls/1_sample_code$ ./testopt -a -
aflag = 1, bflag = 0, cvalue = (null)
Non-option argument -
```

KWANGWOON UNIVERSITY

2019년 1학기 **시스템프로그래밍실습** 4주차

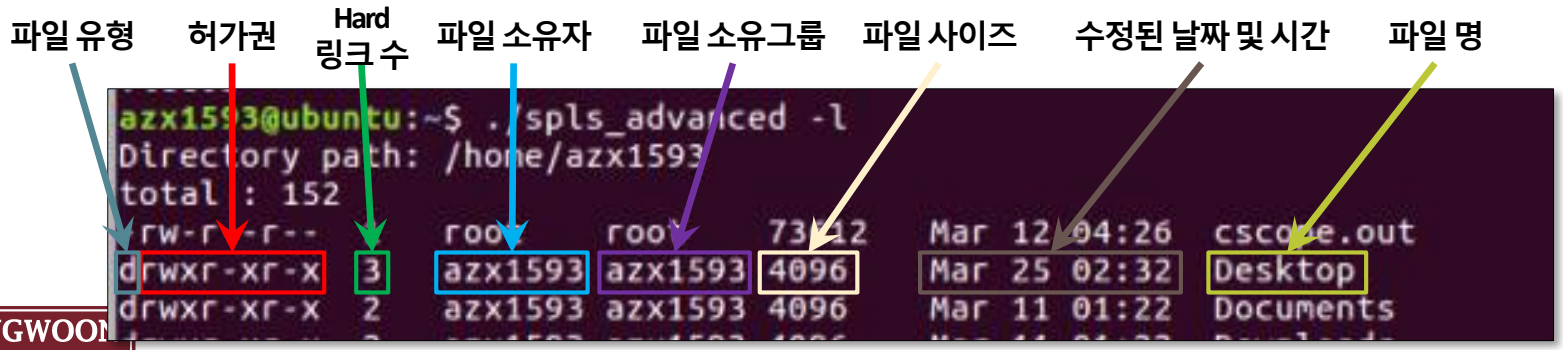
Assignment 2-2

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Assignment 2-2

- **Advanced ls**
  - Assignment 2-2 (“spls_advanced ls”)
    + option implementation (-**a**, -**l**, -**la**)
    + print directory path (e.g. **`Directory path: /home/sslab/splab/`**)
    + print the number of 1K blocks (e.g. **`total 160`**)

  - Requirements
    - Assignment 2-1 과 동일한 조건으로 정렬되어 있어야 함
      - 단, 숨김 파일은 처음 dot('.')을 제외한 문자열이 동일한 조건으로 정렬되어 있어야 함
    - 옵션이 입력된 순서에 무관하게 정상 동작해야 함
    - 파일 혹은 디렉토리 입력 개수에 제한 없음
    - 각 요소(파일 유형, 허가권, 링크 수 등) 간에는 탭을 삽입
    - 기타 요구 사항은 이후의 실행 예제를 참고



Assignment 2-2

- **실행 예제**
  - 존재하지 않는 파일이나 디렉토리가 입력으로 들어온 경우
    - $ ./spls_advanced –la
      - 아래와 같이 에러 메시지 출력

```
azx1593@ubuntu:~/Desktop/sss/ffff$ ./spls_advanced -la
Directory path: /home/azx1593/Desktop/sss/ffff
total : 36
drwxrwxr-x  2   azx1593 azx1593 4096    Mar 28 20:23  .
drwxrwxr-x  5   azx1593 azx1593 4096    Mar 28 20:23  ..
-rwxrwxr-x  1   azx1593 azx1593 24600   Mar 27 01:22  spls_advanced
azx1593@ubuntu:~/Desktop/sss/ffff$ ./spls_advanced -la dkfsji
cannot access dkfsji: No such file or directory
```

      - 단, 존재하지 않는 파일이나 디렉토리와 올바른 파일 혹은 디렉토리 명이 입력으로 같이 들어온 경우,
        - 존재하지 않는 파일 및 디렉토리에 관한 에러 메시지를 상단에 일괄 출력
        - 올바른 파일 및 디렉토리에 대한 결과를 하단에 순서대로 출력

Assignment 2-2

- **실행 예제** (cont'd)
  - $ ./spls_advanced –l

```
azx1593@ubuntu:~$ ./spls_advanced -l
Directory path: /home/azx1593
total : 152
-rw-r--r--  1   root    root    73612   Mar 12 04:26  cscope.out
drwxr-xr-x  3   azx1593 azx1593 4096    Mar 25 02:32  Desktop
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Documents
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Downloads
-rw-r--r--  1   azx1593 azx1593 8980    Mar 11 00:42  examples.desktop
```

  - $ ./spls_advanced –a

```
azx1593@ubuntu:~$ ./spls_advanced -a
.
..
.bashrc
.bash_history
.bash_logout
.cache
.config
cscope.out
Desktop
.dmrc
Documents
Downloads
examples.desktop
.gconf
.gnupg
.ICEauthority
.local
```

Assignment 2-2

- **실행 예제** (cont'd)
  - $ ./spls_advanced -la

```
azx1593@ubuntu:~$ ./spls_advanced -la
Directory path: /home/azx1593
total : 276
drwxr-xr-x  18  azx1593 azx1593 4096    Mar 27 01:26  .
drwxr-xr-x  4   root    root    4096    Mar 11 22:05  ..
-rw-r--r--  1   azx1593 azx1593 3771    Mar 11 00:42  .bashrc
-rw-------  1   azx1593 azx1593 50743   Mar 27 01:28  .bash_history
-rw-r--r--  1   azx1593 azx1593 220     Mar 11 00:42  .bash_logout
drwx------  13  azx1593 azx1593 4096    Mar 25 02:06  .cache
drwx------  15  azx1593 azx1593 4096    Mar 25 02:14  .config
-rw-r--r--  1   root    root    73612   Mar 12 04:26  cscope.out
drwxr-xr-x  3   azx1593 azx1593 4096    Mar 25 02:32  Desktop
-rw-r--r--  1   azx1593 azx1593 25      Mar 11 01:22  .dmrc
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Documents
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Downloads
-rw-r--r--  1   azx1593 azx1593 8980    Mar 11 00:42  examples.desktop
drwx------  2   azx1593 azx1593 4096    Mar 11 22:07  .gconf
drwx------  3   azx1593 azx1593 4096    Mar 26 23:54  .gnupg
-rw-------  1   azx1593 azx1593 1908    Mar 26 23:54  .ICEauthority
drwx------  3   azx1593 azx1593 4096    Mar 11 01:22  .local
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Music
drwxrwxr-x  2   azx1593 azx1593 4096    Mar 11 18:51  .nano
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Pictures
-rw-r--r--  1   azx1593 azx1593 655     Mar 11 00:42  .profile
drwxrwxr-x  4   azx1593 azx1593 4096    Mar 27 00:33  projects
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Public
-rwxrwxr-x  1   azx1593 azx1593 24600   Mar 27 01:22  spls_advanced
drwxrwxr-x  4   azx1593 azx1593 4096    Mar 13 00:16  sslab
-rw-r--r--  1   azx1593 azx1593 0       Mar 11 01:24  .sudo_as_admin_successful
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Templates
drwxr-xr-x  2   azx1593 azx1593 4096    Mar 11 01:22  Videos
-rw-------  1   azx1593 azx1593 5507    Mar 14 02:33  .viminfo
-rw-------  1   azx1593 azx1593 51      Mar 26 23:54  .Xauthority
-rw-------  1   azx1593 azx1593 82      Mar 26 23:54  .xsession-errors
```

Assignment 2-2

- **실행 예제** (cont'd)
  - $ ./spls_advanced –la /home /var

```
azx1593@ubuntu:~$ ./spls_advanced -la /home /var
Directory path: /home
total : 16
drwxr-xr-x  4   root     root     4096     Mar 11 22:05  .
drwxr-xr-x  24  root     root     4096     Mar 11 06:39  ..
drwxr-xr-x  18  azx1593  azx1593  4096     Mar 27 01:26  azx1593
drwxr-xr-x  15  sp2019000000     sp2019000000     4096     Mar 11 22:09  sp2019000000
Directory path: /var
total : 56
drwxr-xr-x  14  root     root     4096     Jul 30 17:50  .
drwxr-xr-x  24  root     root     4096     Mar 11 06:39  ..
drwxr-xr-x  2   root     root     4096     Mar 26 23:58  backups
drwxr-xr-x  16  root     root     4096     Mar 11 00:45  cache
drwxrwxrwx  2   root     whoopsie          4096     Mar 12 07:35  crash
drwxr-xr-x  67  root     root     4096     Mar 27 00:18  lib
drwxrwxr-x  2   root     staff    4096     Apr 12 13:14  local
drwxrwxrwx  4   root     root     120      Mar 27 00:18  lock
drwxrwxr-x  14  root     syslog   4096     Mar 27 00:18  log
drwxrwxr-x  2   root     mail     4096     Jul 30 17:30  mail
drwxrwxrwx  2   root     whoopsie          4096     Jul 30 17:40  metrics
drwxr-xr-x  2   root     root     4096     Jul 30 17:30  opt
drwxr-xr-x  28  root     root     980      Mar 27 01:28  run
drwxr-xr-x  2   root     root     4096     Jul 19 03:48  snap
drwxr-xr-x  7   root     root     4096     Jul 30 17:37  spool
drwxrwxrwx  21  root     root     4096     Mar 27 01:28  tmp
```

Assignment 2-2

- **실행 예제** (cont'd)
  - $ ./spls_advanced Makefile ls.c

```
azx1593@ubuntu:~/Desktop/Untitled Folder$ ./spls_advanced Makefile ls.c
ls.c
Makefile
```

  - ./spls_advanced

```
azx1593@ubuntu:~/Desktop/Untitled Folder$ ./spls_advanced
gg
hello1.c
hello2.c
hello2.h
ls.c
ls.c'
ls1.c
Makefile
run
spls_advanced
t.c
tags
test
TT
```

Assignment 2-2

- **Softcopy Upload**
  - 제출 파일
    - 보고서
    - 요일_학번_ls.c , 요일 _학번_ls(Makefile)
      - e.g. fri_20190000_ls.c, fri_20190000_ls(Makefile)
    - Soruce code copy 적발 시 예외 없이 0점 처리
  - 위 파일들을 압축해서 제출 (파일명: 실습 요일_2-2_학번.tar.gz)
    - e.g. 월1,2 ➔ **mon**_2-2_2017202000.**tar.gz**
    - e.g. 화3,4 ➔ **tue**_2-2_2017202000.**tar.gz**
    - e.g. 금5,6 ➔ **fri**_2-2_2017202000.**tar.gz**
  - U-Campus의 과제 제출에 4월 12일(금) 23:59:59까지 제출
    - U-Campus에 올린 후 다시 다운로드 받아서 파일이 정확한지 확인
  - 미리 공지한 바와 같이, delay 받지 않음 (예외 없음)
  - Ubuntu 16.04 64bits 환경에서 채점

- **과제 질문 관련**
  - 해당 과제 출제 담당 조교에게 이메일로 문의 ➔ **조수익 조교 (azx1593@kw.ac.kr)**
  - 과제 제출 마감 당일에는 오후 4시까지 도착한 질문 메일에만 답변

Assignment 2-2

- **표지**
  - 다음의 내용은 필히 기록
    - 과제 이름 (e.g. Assignment#2-2)
    - 분반 (요일, 담당 교수님)
    - 본인 인적 사항 (학번, 이름)

- **과제 내용**
  - Introduction : 5줄 이하
  - Flow chart : 4주차 자료의 Appendix 참고
  - Pseudo code : 4주차 자료의 Appendix 참고
  - Result : 수행한 내용을 캡쳐 이미지와 함께 설명
  - Conclusion : 결론 및 고찰

- **보고서 이름은 "실습 요일_과제명_학번"으로 수정**
  - e.g. 월1,2 ➔ **mon**_2-2_2017202000.**pdf**
  - e.g. 화3,4 ➔ **tue**_2-2_2017202000.**pdf**
  - e.g. 금5,6 ➔ **fri**_2-2_2017202000.**pdf**