KWANGWOON UNIVERSITY

2023년 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

```
sslab@ubuntu:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

# 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

```
sslab@ubuntu:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,sslab
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
```

# **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(test.c)**

```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 varialve==========//
                printf("optarg = %s\t optind = %d\t opterr= %d\t optopt= %c\n", optarg, optind, opterr, optopt);
                //=============================================//

                switch(c)
                {
                        case 'a':                       //just flag on
                                aflag++;
                                break;
                        case 'b':                       // just flag on
                                bflag++;
                                break;
                        case 'c':                       // get argument
                                cvalue = optarg;
                                break;
                        case 'd':                       // opterr ogg
                                opterr = 0;
                                break;
                        case '?':                       // print the error mesage
                                printf("Unknow option character\n");
                                break;

                }


        }

        //=========print flags and c's argument=========//
        printf("\n aflag = %d\t bflag = %d\t cvalue = %s\n", aflag, bflag, cvalue);
        return 0;

}
```

# Example (cont'd)

- **Execution**

```
sslab@ubuntu:~/2_ls$ ./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
Unknow option character
./test: invalid option -- 'e'
optarg = (null)  optind = 4      opterr= 1       optopt= e
Unknow option character
./test: invalid option -- 'l'
optarg = (null)  optind = 4      opterr= 1       optopt= l
Unknow option character
./test: invalid option -- 'l'
optarg = (null)  optind = 4      opterr= 1       optopt= l
Unknow option character
./test: invalid option -- 'o'
optarg = (null)  optind = 5      opterr= 1       optopt= o
Unknow option character

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

# Example

- **Execution** (cont'd)

```
sslab@ubuntu:~/2_ls$ ./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
Unknow option character
optarg = (null)  optind = 5      opterr= 0       optopt= e
Unknow option character
optarg = (null)  optind = 5      opterr= 0       optopt= l
Unknow option character
optarg = (null)  optind = 5      opterr= 0       optopt= l
Unknow option character
optarg = (null)  optind = 6      opterr= 0       optopt= o
Unknow option character

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

```
sslab@ubuntu:~/2_ls$ ./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 c = 0;
        opterr = 0;

        Blank

        printf("aflag = %d\t bflag = %d\t cvalue = %s\n", aflag, bflag, cvalue);

        Blank

        return 0;

}
```

# Lab. (cont'd)

- **Execution**

```
user1@kwadmin-A6000:~/work$ ./testopt
aflag = 0         bflag = 0         cvalue = (null)
user1@kwadmin-A6000:~/work$ ./testopt -a -b
aflag = 1         bflag = 1         cvalue = (null)
user1@kwadmin-A6000:~/work$ ./testopt -ab
aflag = 1         bflag = 1         cvalue = (null)
user1@kwadmin-A6000:~/work$ ./testopt -c foo
aflag = 0         bflag = 0         cvalue = foo
user1@kwadmin-A6000:~/work$ ./testopt -cfoo
aflag = 0         bflag = 0         cvalue = foo
user1@kwadmin-A6000:~/work$ ./testopt arg1
aflag = 0         bflag = 0         cvalue = (null)
Non option argument arg1
user1@kwadmin-A6000:~/work$ ./testopt -d
unknown option character
aflag = 0         bflag = 0         cvalue = (null)
```