2023년 1학기 **시스템프로그래밍실습** 4주차

ls basics

System Software Laboratory
College of Software and Convergence
Kwangwoon Univ.

Contents

gcc

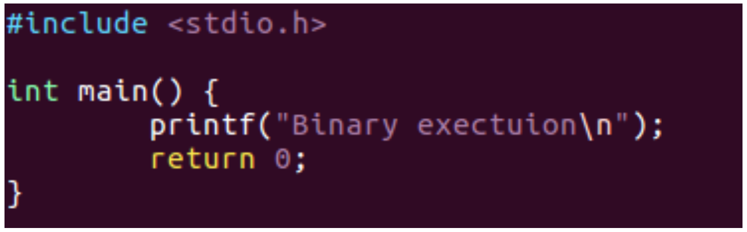
- **GNU Compiler Collection**
- **Options**
  - c : compile or assemble. but do not link
  - o file : Place output in file
    - If not specified, *a.out* by default
- **Example**
  - gcc -c file.c
    ➔ file.o generated
  - gcc -o binary file1.c file2.c
    ➔ executable file (i.e. *binary*) generated

KWANGWOON UNIVERSITY

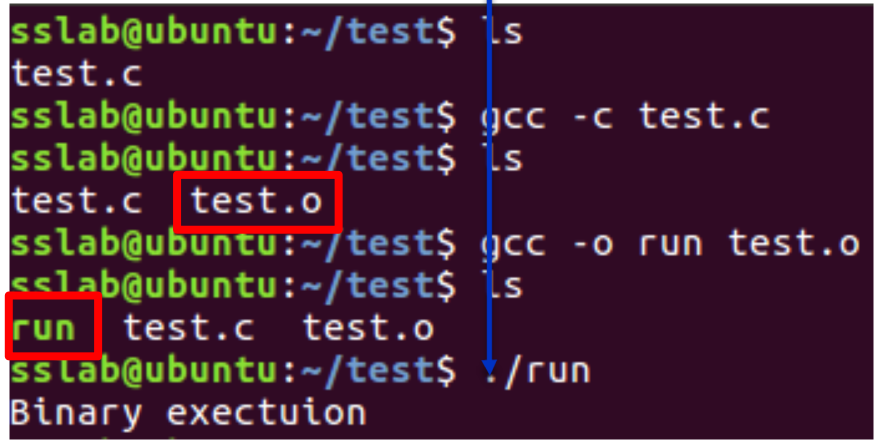
Binary execution

- **Whenever execute program, you must specify the path**
  - .　　: current directory
  - ..　　: parent directory

- **You can execute program like this**
  - ./binary
    - e.g. "test.c"



```
#include <stdio.h>

int main() {
        printf("Binary exectuion\n");
        return 0;
}
```

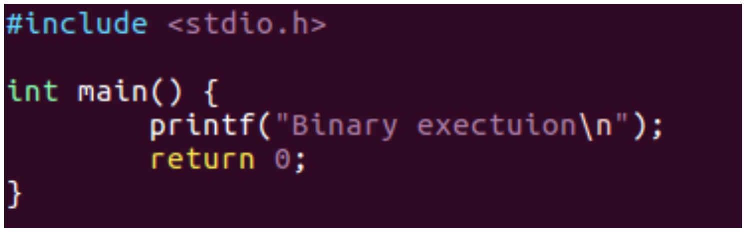


```
sslab@ubuntu:~/test$ ls
test.c
sslab@ubuntu:~/test$ gcc -c test.c
sslab@ubuntu:~/test$ ls
test.c  test.o
sslab@ubuntu:~/test$ gcc -o run test.o
sslab@ubuntu:~/test$ ls
run  test.c  test.o
sslab@ubuntu:~/test$ ./run
Binary exectuion
```

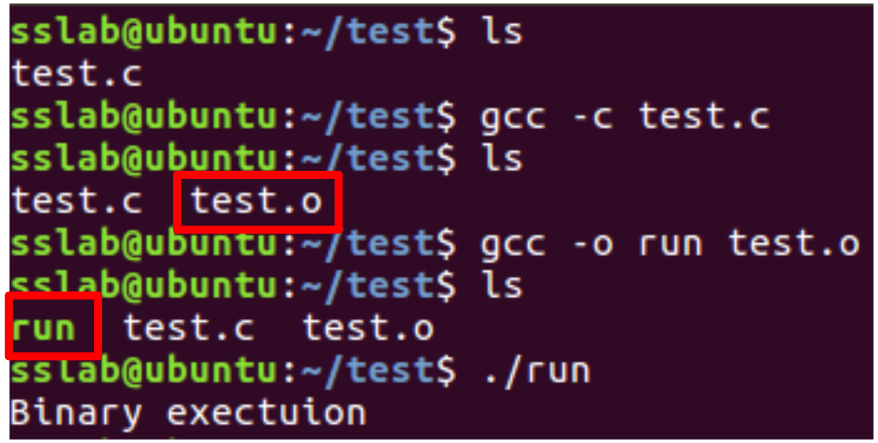
Binary execution

- **Whenever execute program, you must specify the path**
  - .   : current directory
  - ..   : parent directory

- **You can execute program like this**
  - ./binary
    - e.g. “test.c”



```
#include <stdio.h>

int main() {
        printf("Binary exectuion\n");
        return 0;
}
```



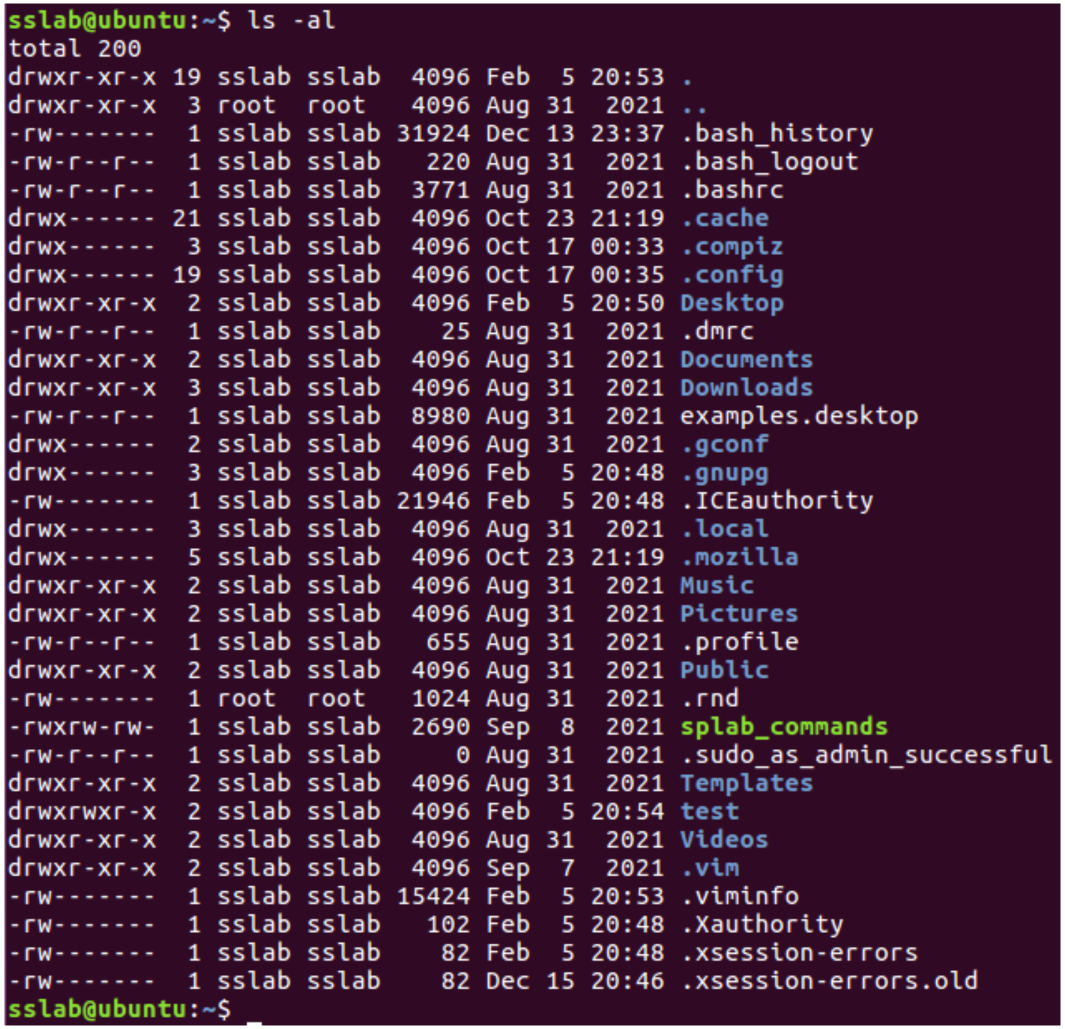
```
sslab@ubuntu:~/test$ ls
test.c
sslab@ubuntu:~/test$ gcc -c test.c
sslab@ubuntu:~/test$ ls
test.c  test.o
sslab@ubuntu:~/test$ gcc -o run test.o
sslab@ubuntu:~/test$ ls
run  test.c  test.o
sslab@ubuntu:~/test$ ./run
Binary exectuion
```

ls – list directory contents

- **Synopsis**
  - $ ls [OPTION]... [FILE]...
- **Description**
  - List information about the FILEs (current directory by default)
- **Options**
  - -a
    - do not ignore entries starting with . (i.e. *hidden file or directory*)
  - -l
    - use a long listing format
    - File mode, number of links, owner name, group name, number of bytes in the file, abbreviated month, day-of-month file was last modified, hour file last modified, minute file last modified, and the pathname

ls – list directory contents (cont'd)

- **Example**



```
sslab@ubuntu:~$ ls -al
total 200
drwxr-xr-x 19 sslab sslab  4096 Feb  5 20:53 .
drwxr-xr-x  3 root  root   4096 Aug 31  2021 ..
-rw-------  1 sslab sslab 31924 Dec 13 23:37 .bash_history
-rw-r--r--  1 sslab sslab   220 Aug 31  2021 .bash_logout
-rw-r--r--  1 sslab sslab  3771 Aug 31  2021 .bashrc
drwx------ 21 sslab sslab  4096 Oct 23 21:19 .cache
drwx------  3 sslab sslab  4096 Oct 17 00:33 .compiz
drwx------ 19 sslab sslab  4096 Oct 17 00:35 .config
drwxr-xr-x  2 sslab sslab  4096 Feb  5 20:50 Desktop
-rw-r--r--  1 sslab sslab    25 Aug 31  2021 .dmrc
drwxr-xr-x  2 sslab sslab  4096 Aug 31  2021 Documents
drwxr-xr-x  3 sslab sslab  4096 Aug 31  2021 Downloads
-rw-r--r--  1 sslab sslab  8980 Aug 31  2021 examples.desktop
drwx------  2 sslab sslab  4096 Aug 31  2021 .gconf
drwx------  3 sslab sslab  4096 Feb  5 20:48 .gnupg
-rw-------  1 sslab sslab 21946 Feb  5 20:48 .ICEauthority
drwx------  3 sslab sslab  4096 Aug 31  2021 .local
drwx------  5 sslab sslab  4096 Oct 23 21:19 .mozilla
drwxr-xr-x  2 sslab sslab  4096 Aug 31  2021 Music
drwxr-xr-x  2 sslab sslab  4096 Aug 31  2021 Pictures
-rw-r--r--  1 sslab sslab   655 Aug 31  2021 .profile
drwxr-xr-x  2 sslab sslab  4096 Aug 31  2021 Public
-rw-------  1 root  root   1024 Aug 31  2021 .rnd
-rwxrw-rw-  1 sslab sslab  2690 Sep  8  2021 splab_commands
-rw-r--r--  1 sslab sslab     0 Aug 31  2021 .sudo_as_admin_successful
drwxr-xr-x  2 sslab sslab  4096 Aug 31  2021 Templates
drwxrwxr-x  2 sslab sslab  4096 Feb  5 20:54 test
drwxr-xr-x  2 sslab sslab  4096 Aug 31  2021 Videos
drwxr-xr-x  2 sslab sslab  4096 Sep  7  2021 .vim
-rw-------  1 sslab sslab 15424 Feb  5 20:53 .viminfo
-rw-------  1 sslab sslab   102 Feb  5 20:48 .Xauthority
-rw-------  1 sslab sslab    82 Feb  5 20:48 .xsession-errors
-rw-------  1 sslab sslab    82 Dec 15 20:46 .xsession-errors.old
sslab@ubuntu:~$
```

ls – list directory contents (cont'd)

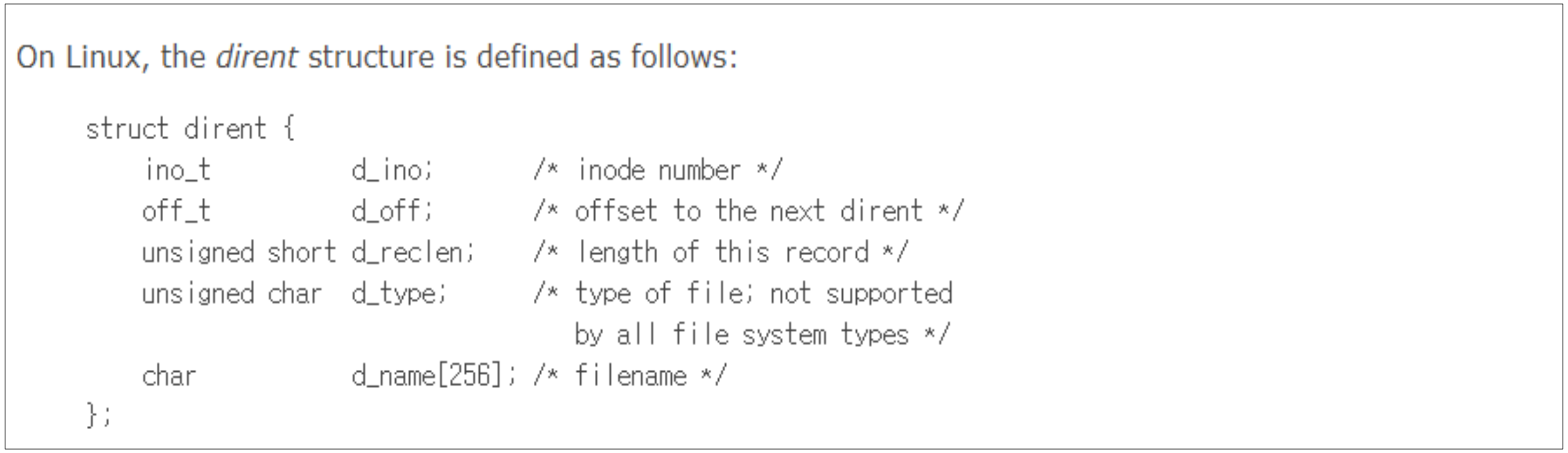
- **Data types**
  - **typedef struct __distream DIR**
  - **struct dirent**
  - struct passwd
  - struct stat
  - struct tm
- **System Calls & Functions**
  - **opendir(), readdir(), closedir()**
  - stat()
  - getgrgid(), getpwuid()
  - localtime()
  - getwd()
  - getopt()
  - fnmatch()

Data types

- **header : <dirent.h>**
- **Data type : typedef struct __dirstream DIR**
  - The DIR data type represents a directory stream

Data types (cont'd)

- **header : <dirent.h>**
- **Data type : struct dirent**
- **Members :**
  - __ino_t d_ino // inode number
  - char d_name[256] // filename



On Linux, the *dirent* structure is defined as follows:

```
struct dirent {
    ino_t          d_ino;       /* inode number */
    off_t          d_off;       /* offset to the next dirent */
    unsigned short d_reclen;    /* length of this record */
    unsigned char  d_type;      /* type of file; not supported
                                   by all file system types */
    char           d_name[256]; /* filename */
};
```

Source : https://linux.die.net/man/3/readdir_r

Reading directories

- **#include <dirent.h>**
  DIR *opendir(const char *name);
  - opens a directory stream corresponding to the directory **name**
  - returns a pointer to the directory stream (on error, NULL is returned)
  - The stream is positioned at the first entry in the directory

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

struct dirent *readdir(DIR *dp);
```

KWANGWOON UNIVERSITY

Reading directories (cont'd)

- **#include <dirent.h>**

  struct dirent *readdir(DIR *dirp);
  - returns a pointer to a dirent structure representing the next directory entry in the directory stream pointed to by **dirp**
  - It returns NULL on reaching the end of the directory steam or if an error occurred.
  - If an error occurs, NULL is returned and **errno** is set appropriately.

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

struct dirent *readdir(DIR *dp);
```

KWANGWOON UNIVERSITY

Reading directories (cont'd)

- **#include <dirent.h>**

  int closedir(DIR *dirp);
  - closes the directory stream associated with **dirp**
  - The directory stream descriptor **dirp** is not available after this call.
  - returns 0 on success. On error, -1 is returned

```
int closedir(DIR *dp);
```

KWANGWOON UNIVERSITY

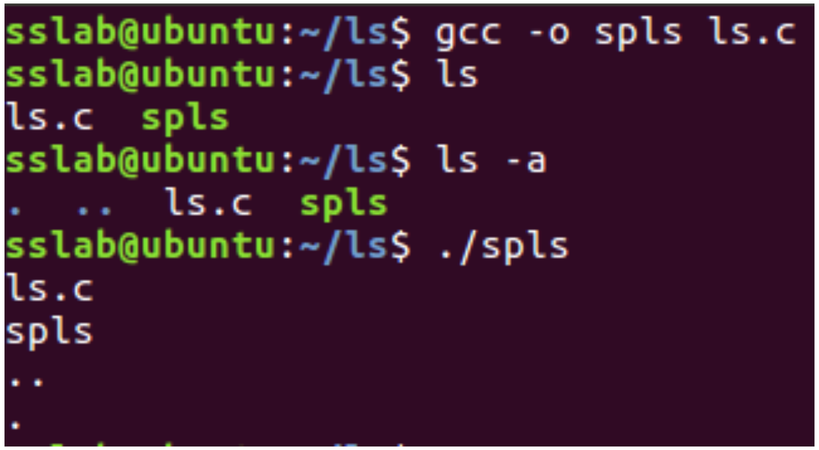
실습 이해를 돕기위한 예시

- 소스 코드

```
#include <stdio.h>
#include <dirent.h>

int main(){
        DIR *dirp;
        struct  dirent *dir;

        ____ = opendir(".");

        while(( ______________ )!=NULL){
                printf("%s\n", _______ );
        }

        closedir( ____ );

        return 0;
}
```

$ vi ls.c

- 결과 화면



```
sslab@ubuntu:~/ls$ gcc -o spls ls.c
sslab@ubuntu:~/ls$ ls
ls.c  spls
sslab@ubuntu:~/ls$ ls -a
.  ..  ls.c  spls
sslab@ubuntu:~/ls$ ./spls
ls.c
spls
..
.
```

$ ./spls