# Command

*** 명령어**

| 크기가 0인 파일을 생성 | touch <filename> |
|---|---|

touch memo.txt

touch test.c

확장자를 본인이 지정해야 함

| 확장자가 c인 파일을 컴파일 | gcc -o <progam_name> <c_file_name> |
|---|---|

1. gcc test.c : a.out 이라는 실행파일이 생성됨
2. gcc -o test test.c : test라는 실행파일이 생성됨
3. gcc -o my_prog test.c : my_prog라는 실행파일이 생성됨

   2, 3의 경우 test.c라는 코드를 컴파일 했기 때문에 결과는 같음

   단, 결과로 나오는 실행파일의 이름만 달라짐

   즉, 내용물은 뒤의 c파일의 영향을 받음

   일반적인 경우 2번과 같이 실행파일명과 c파일명을 일치시켜주는 경우가 많음

| 현재 디렉터리 확인 | pwd |
|---|---|

```
$ pwd
/home/runner/Cstudy
```

gcc를 하기위해서는 c파일이 있는 위치에서 진행하여야함

| 디렉터리 내용 확인 | ls <option> |
|---|---|

```
$ ls
main.c  Makefile  replit.nix
```

따라서, ls 명령어를 통해 해당 c파일이 현재 위치에 있는지 확인

```
$ ls -asl
total 20
0 drwxr-xr-x 1 runner runner  120 Feb 14 09:08 .
0 drwxrwxrwx 1 runner runner   64 Feb 14 08:56 ..
4 -rw-r--r-- 1 runner runner  173 Feb 14 09:08 .breakpoints
0 drwxr-xr-x 1 runner runner   12 Oct 12 20:07 .cache
0 drwxr-x--- 1 runner runner  452 Feb 14 08:56 .ccls-cache
4 -rw-r--r-- 1 runner runner   77 Jan 19 00:12 main.c
4 -rw-r--r-- 1 runner runner  250 Jan 19 00:09 Makefile
4 -rw-r--r-- 1 runner runner 1274 Jan 19 00:10 .replit
4 -rw-r--r-- 1 runner runner   81 Jan 18 23:59 replit.nix
```

구체적인 정보를 확인하기 위해 -asl 옵션을 사용할 수 도 있음

여기에서 . 은 현재 디렉터리, .. 은 부모 디렉터리를 의미

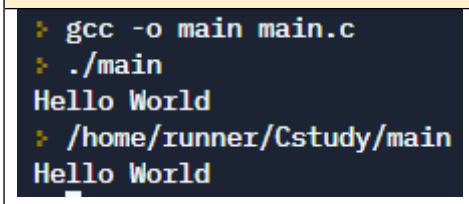
| 디렉터리 이동 | cd |
|---|---|

cd : 홈 디렉터리로 이동

cd <전체경로명> : 해당하는 경로로 이동 - cd /home/runner/Cstudy

cd 디렉터리명 : 현재 디렉터리내에 존재하는 하위 디렉터리로 이동 - cd week01

cd .. : 부모 디렉터리로 이동 - /home/runner/Cstudy (현재위치), /home/runner/ (부모위치)

| 프로그램의 실행 | <경로명> |
| --- | --- |
|  <br>대부분의 리눅스 시스템은 경로명을 입력하여 프로그램을 실행<br>번거로움을 피하기 위해 .(현재디렉터리)를 사용<br>원래 /home/runner/Cstudy/main과 같이 전체경로명을 입력해야함<br>이를 ./main으로 대체 즉, .은 /home/runner/Cstudy를 의미 | |
| 이외의 유용한 명령어 | |
| 빈 디렉터리 생성 | mkdir <dir_name> |
| 빈 디렉터리 삭제 | rmdir <dir_name> |
| 파일 삭제 | rm <file_name> |
| 터미널(콘솔화면) 정리 | clear |