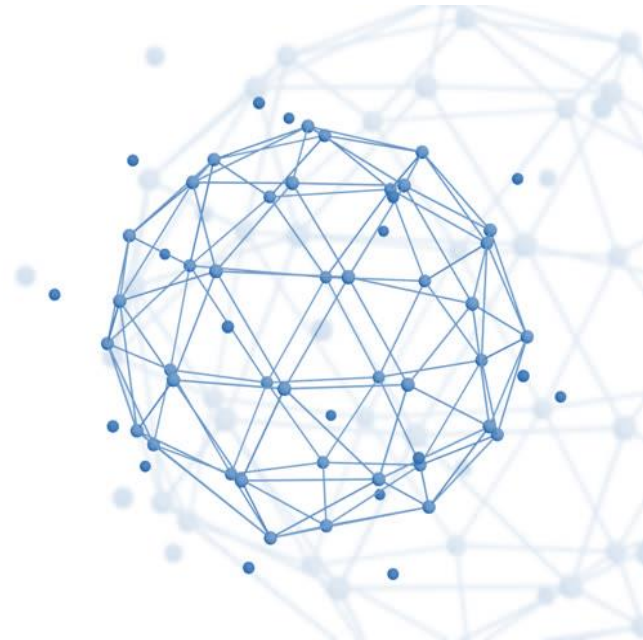


Vi Command & GCC

2021.03.18

Sang-woo Lee

glutton.leesw@gmail.com



Contents

- Vi
- GCC



Vi (Visual editor)

- Vi editor
 - 유닉스 환경에서 가장 많이 쓰이는 텍스트 편집기

A screenshot of a terminal window titled "2 Ubuntu". The terminal displays the following C code:

```
#include <stdio.h>

int main(void)
{
    printf("Hello world!\n");
    return 0;
}

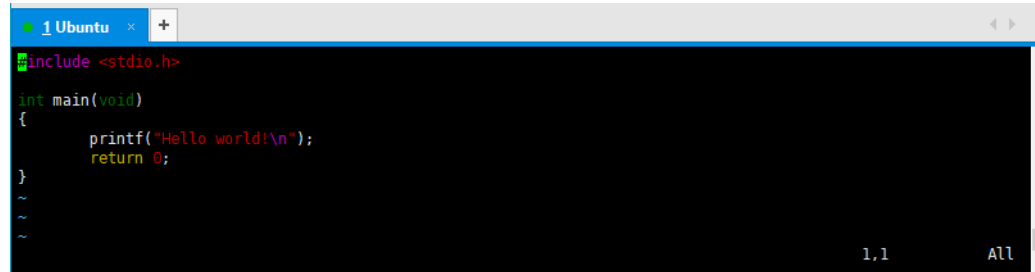
~
~
~
~
~
~
~
~
~
```

The cursor is at the end of the last tilde (~). At the bottom right of the terminal area, it says "1,1 All". Below the terminal area, there is a status bar with the text "현재 탭으로만 텍스트 보내기" and a button icon. At the very bottom, there is a system tray area with icons for SSH2, xterm, and session information: "현재 세션의 등록 정보 대화 상자를 엽니다.", "SSH2", "xterm", "112x23", "1,1", "2 세션", and "CAP NUM".

Vi (Visual editor)

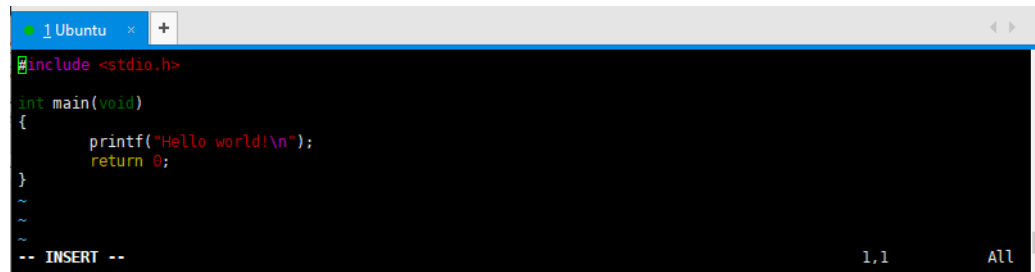
- Vi 모드

- 명령 모드 : ESC 키, vi 화면 좌측 하단에 아무런 표시가 없는 상태



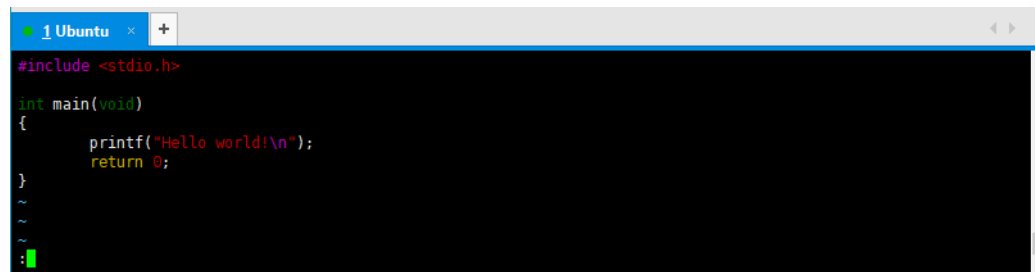
```
1 Ubuntu x +
#include <stdio.h>
int main(void)
{
    printf("Hello world!\\n");
    return 0;
}
~
~
~
1.1 All
```

- 입력 모드 : i 키, vi 화면 좌측 하단에 -- INSERT -- 표시



```
1 Ubuntu x +
#include <stdio.h>
int main(void)
{
    printf("Hello world!\\n");
    return 0;
}
~
~
~
-- INSERT --
1.1 All
```

- Ex 모드 : shift + : (콜론) , vi 화면 좌측 하단에 : 표시

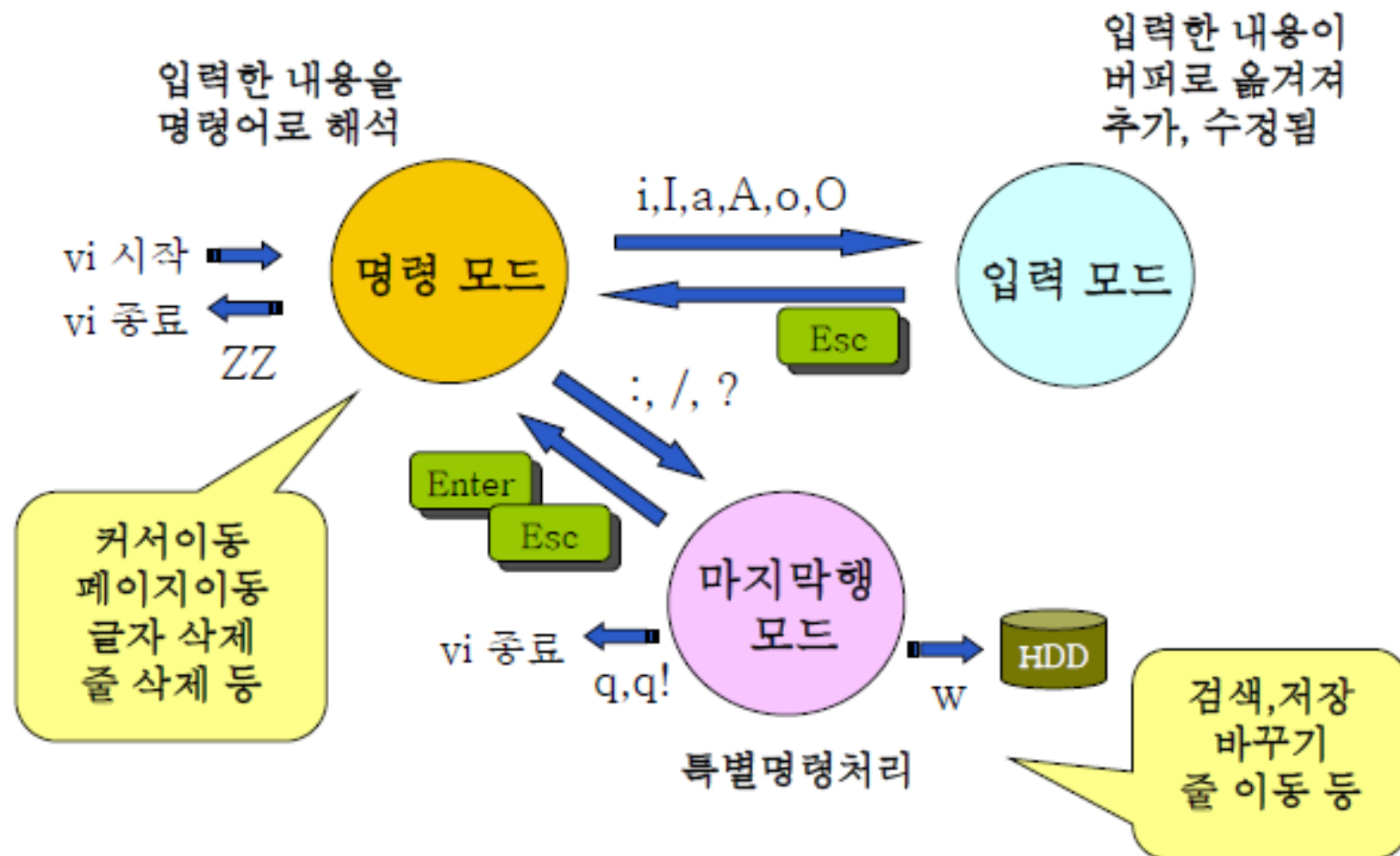


```
1 Ubuntu x +
#include <stdio.h>
int main(void)
{
    printf("Hello world!\\n");
    return 0;
}
~
~
~
:
1.1 All
```



Vi (Visual editor)

- Vi 모드



Vi (Visual editor)

- 입력 명령 (명령 모드 → 입력 모드)
 - 입력이 끝나면 **입력 모드**에서 **명령 모드**로 돌아와야 함
 - ESC 키를 통해 입력모드에서 명령모드로 전환
- 문자, 행 삽입 명령어

명령어	작업 설명
i	커서 왼쪽에 문자 삽입
a	커서 오른쪽에 문자 삽입
I	커서 왼쪽, 행의 처음에 문자 삽입
A	커서 오른쪽, 행의 끝에 문자 삽입
o	커서 아래에 행 삽입
O	커서 위에 행 삽입

Vi (Visual editor)

- 커서 이동 명령어
 - 현재 화면에서 커서 이동

명령어	작업 설명
h	왼쪽으로 이동
j	아래로 이동
k	위로 이동
l	오른쪽으로 이동
Shift + \$	현재 라인의 가장 끝으로 이동
Shift + ^	현재 라인의 가장 앞으로 이동

Vi (Visual editor)

- 커서 이동 명령어
 - 지정한 곳으로 이동

명령어	작업 설명
:n / nG	n번째 줄로 커서 이동
:\$ / G	파일의 마지막 행으로 가기
n+	n줄 만큼 앞으로 이동
n-	n줄 만큼 뒤로 이동
(현재 문장의 처음으로 이동
)	다음 문장의 처음으로 이동
{	현재 문단의 처음으로 이동
}	다음 문단의 처음으로 이동

Vi (Visual editor)

- 편집 명령어

명령어	작업 설명
Shift + v	Visual line (여러 줄 선택)
y	복사
yy	한 줄 복사
p	붙이기
d	삭제
dd	한 줄 삭제
Shift + d	커서 위치부터 행 끝까지 삭제
u	이전 수행 명령 취소
U	해당 줄의 모든 편집 취소

Vi (Visual editor)

- 저장 명령어

- 저장하거나 종료하려면 명령 모드로 돌아와야 함

명령어	작업 설명
:w	현재의 파일명으로 파일 저장
:w 파일명	지정한 파일명으로 파일 저장

- 종료 명령어

- 종료 또는 저장 후 종료

명령어	작업 설명
:q	작업 내용을 저장하였으면 vi 종료
:q!	작업 내용을 저장하지 않고 vi 종료
:wq	작업 내용을 저장한 후 vi 종료
:wq 파일명	작업 내용을 지정한 파일명으로 저장한 후 vi 종료
zz	작업 내용을 저장한 후 vi 종료

Vi (Visual editor)

- 문자열 검색

명령어	작업 설명
/찾는 문자열	아래쪽으로 찾는 문자열 검색
?찾는 문자열	위쪽으로 찾는 문자열 검색
n	문자열의 다음으로 계속 검색 (순방향)
N	문자열의 이전으로 계속 검색 (역방향)

- 찾기/바꾸기

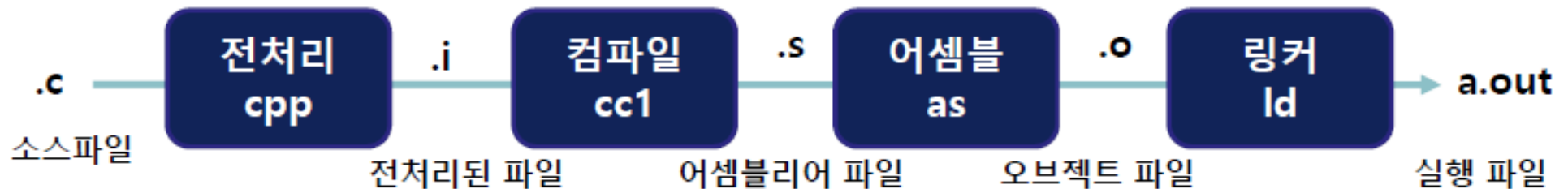
명령어	작업 설명
:s/찾을 문자열/치환 문자열	현재 행의 찾을 문자열을 치환 문자열로 대체

- 행 번호 표시

명령어	작업 설명
:set nu	행 번호 표시
:set nonu	행 번호 숨기기

GCC (GNU Compiler Collection)

- GCC
 - GNU 프로젝트 C와 C++ 컴파일러
- GCC가 실행시키는 프로그램
 - cpp : 전처리기
 - cc1 : 컴파일러
 - as : 어셈블러
 - ld : 링커



GCC (GNU Compiler Collection)

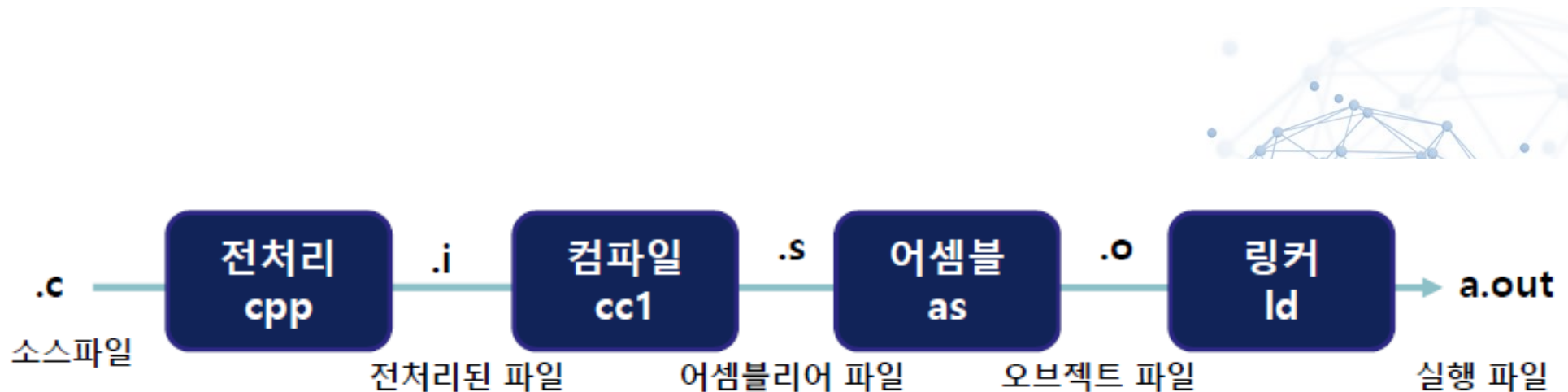
- 전처리 단계
 - `#include` , `#define` 등의 전처리 부분을 처리
 - `.i` 확장자를 가진 파일 생성
- 컴파일 단계
 - 전처리된 파일을 가지고 어셈블리어로 된 파일을 생성
 - `.s` 확장자를 가진 파일 생성
- 어셈블 단계
 - 어셈블리어로 된 파일을 기계가 직접 이해할 수 있는 기계어로 된 오브젝트 파일 생성
 - `.o` 확장자를 가진 파일 생성
- 링크 단계
 - `printf` , `scanf`와 같은 라이브러리 함수와 오브젝트 파일들을 연결하여 실행 파일로 생성



GCC (GNU Compiler Collection)

- GCC 옵션

옵션	작업 설명
-E	전처리만 실행, 컴파일과 어셈블, 링크 수행 안함
-S	컴파일만 실행, 어셈블과 링크 수행 함 (순방향)
-c	소스 파일을 컴파일과 어셈블 수행, 링크를 수행 안함
-o	바이너리 형식의 출력 파일 이름을 지정 (실행 파일명)



GCC (GNU Compiler Collection)

- GCC 옵션 실습
 - hello_os.c 소스 코드 작성

```
#include <stdio.h>

int main(void)
{
    printf("Hello Operating System!\n");
    return 0;
}
```

GCC (GNU Compiler Collection)

- GCC 옵션 실습
 - -E 옵션 사용 (gcc -E hello_os.c)

```
extern char *ctermid (char *__s) __attribute__ ((__nothrow__ , __leaf__));
# 913 "/usr/include/stdio.h" 3 4
extern void flockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));

extern int ftrylockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));

extern void funlockfile (FILE *__stream) __attribute__ ((__nothrow__ , __leaf__));
# 943 "/usr/include/stdio.h" 3 4

# 2 "hello_os.c" 2

int main(void)
{
    printf("Hello Operating System!\n");
    return 0;
}
solbee@solbee:~$
```


GCC (GNU Compiler Collection)

- GCC 옵션 실습
 - -E 옵션 사용 (gcc -E hello_os.c -o hello_os.i)

```
solbee@solbee:~$ gcc -E hello_os.c -o hello_os.i
solbee@solbee:~$ ls -l
total 32
-rw-rw-r-- 1 solbee solbee   88 Feb 12 16:09 hello.c
-rw-rw-r-- 1 solbee solbee   88 Feb 12 16:10 hello_os.c
-rw-rw-r-- 1 solbee solbee 17596 Feb 12 16:19 hello_os.i
-rw-rw-r-- 1 solbee solbee   506 Feb 12 16:18 hello_os.s
solbee@solbee:~$
```



GCC (GNU Compiler Collection)

- GCC 옵션 실습
 - -S 옵션 사용 (gcc -S hello_os.c)

```
solbee@solbee:~$ gcc -S hello_os.c
solbee@solbee:~$ ls -l
total 32
-rw-rw-r-- 1 solbee solbee  88 Feb 12 16:09 hello.c
-rw-rw-r-- 1 solbee solbee  88 Feb 12 16:10 hello_os.c
-rw-rw-r-- 1 solbee solbee 17596 Feb 12 16:17 hello_os.i
-rw-rw-r-- 1 solbee solbee  506 Feb 12 16:18 hello_os.s
solbee@solbee:~$
```



GCC (GNU Compiler Collection)

- GCC 옵션 실습
 - -c 옵션 사용 (gcc -c hello_os.c)

```
solbee@solbee:~$ gcc -c hello_os.c
solbee@solbee:~$ ls -l
total 36
-rw-rw-r-- 1 solbee solbee  88 Feb 12 16:09 hello.c
-rw-rw-r-- 1 solbee solbee  88 Feb 12 16:10 hello_os.c
-rw-rw-r-- 1 solbee solbee 17596 Feb 12 16:19 hello_os.i
-rw-rw-r-- 1 solbee solbee  1044 Feb 12 16:20 hello_os.o
-rw-rw-r-- 1 solbee solbee   506 Feb 12 16:18 hello_os.s
solbee@solbee:~$
```

GCC (GNU Compiler Collection)

- GCC 옵션 실습
 - -c 옵션 사용 (gcc hello_os.c -o hello_os)

```
solbee@solbee:~$ gcc hello_os.o -o hello_os
solbee@solbee:~$ ls -l
total 44
-rw-rw-r-- 1 solbee solbee  88 Feb 12 16:09 hello.c
-rwxrwxr-x 1 solbee solbee 7303 Feb 12 16:23 hello_os
-rw-rw-r-- 1 solbee solbee  88 Feb 12 16:10 hello_os.c
-rw-rw-r-- 1 solbee solbee 17596 Feb 12 16:19 hello_os.i
-rw-rw-r-- 1 solbee solbee 1044 Feb 12 16:20 hello_os.o
-rw-rw-r-- 1 solbee solbee  506 Feb 12 16:18 hello_os.s
solbee@solbee:~$ ./hello_os
Hello Operating System!
solbee@solbee:~$
```

GCC (GNU Compiler Collection)

- 여러 개의 오브젝트 파일을 합쳐서 실행 파일 만들기
 - main.c와 print.c 소스 코드 작성

```
#include <stdio.h>

extern void print();

int main(){
    print();
}
```

```
#include <stdio.h>

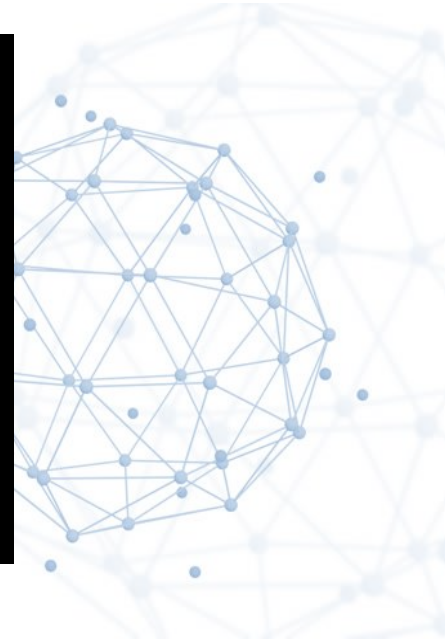
void print(){
    print("Operating System GCC Test!\n");
}
```

GCC (GNU Compiler Collection)

- 여러 개의 오브젝트 파일을 합쳐서 실행 파일 만들기
 - 컴파일 수행 (gcc main.c print.c -o gcc_test)

```
solbee@solbee:~$ gcc main.c -o gcc_test
/tmp/ccCDiaAg.o: In function `main':
main.c:(.text+0x7): undefined reference to `print'
collect2: error: ld returned 1 exit status
solbee@solbee:~$
```

```
solbee@solbee:~$ gcc main.c print.c -o gcc_test
solbee@solbee:~$ ls -l
total 60
-rwxrwxr-x 1 solbee solbee 7345 Feb 12 16:31 gcc_test
-rw-rw-r-- 1 solbee solbee 88 Feb 12 16:09 hello.c
-rwxrwxr-x 1 solbee solbee 7303 Feb 12 16:23 hello_os
-rw-rw-r-- 1 solbee solbee 88 Feb 12 16:10 hello_os.c
-rw-rw-r-- 1 solbee solbee 17596 Feb 12 16:19 hello_os.i
-rw-rw-r-- 1 solbee solbee 1044 Feb 12 16:20 hello_os.o
-rw-rw-r-- 1 solbee solbee 506 Feb 12 16:18 hello_os.s
-rw-rw-r-- 1 solbee solbee 66 Feb 12 16:25 main.c
-rw-rw-r-- 1 solbee solbee 77 Feb 12 16:30 print.c
solbee@solbee:~$
```



차주 수업 내용

- GCC 및 make 실습



Thank you

