



gcc 컴파일러

로봇SW 교육원

최상훈(shchoi82@gmail.com)

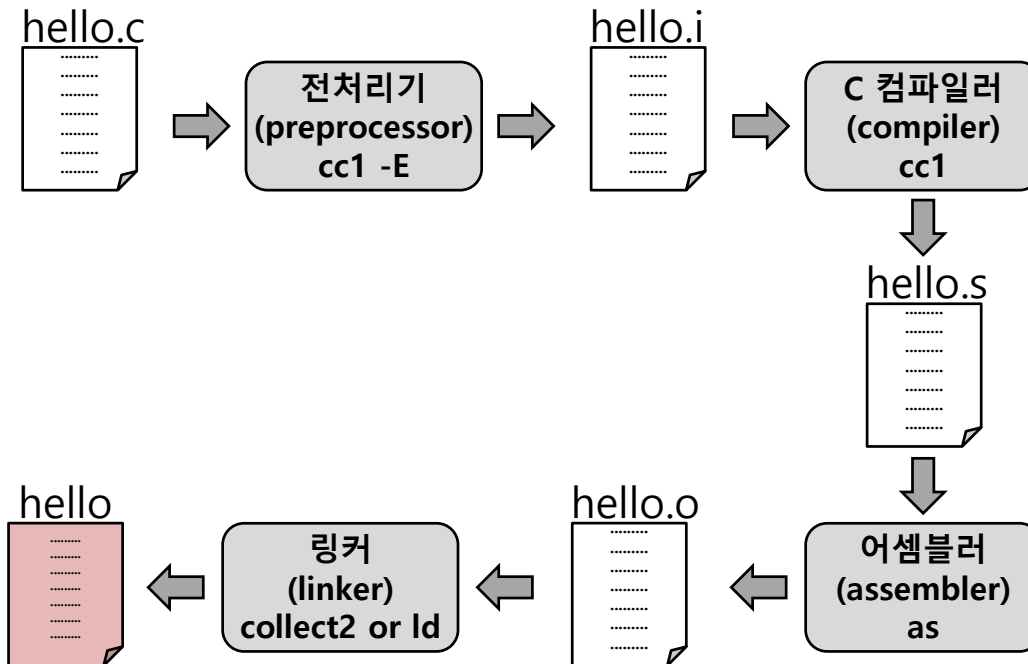
학습 목표

2

- 프로그램 소스 컴파일

gcc 컴파일러

- GCC
 - Gnu C Compiler → GNU Compiler Collection
- gcc 컴파일러 수행 과정



< gcc -Wall -W -o hello hello.c 의 실행과정 >

- -v 옵션과 --save-temps 옵션으로 확인 가능
- driver program

gcc 컴파일러

4

- `--save-temps` : 컴파일 과정에서 생성되는 중간파일 저장함
- `-v` : 컴파일 과정을 화면으로 출력함
- `-o` : 결과 파일명 설정함
- `-Wall` : 모호한 문법에 대한 경고 메시지 출력
- `-W` : 합법적이지만 모호한 코딩에 대한 부가적인 정보 제공
- `-c` : .o 오브젝트파일 생성

ex) `gcc -v --save-temps -o helloworld helloworld.c`

ex) `gcc -Wall -W -o helloworld helloworld.c`

ex) `gcc -Wall -W -c helloworld.c`

실습 1: helloworld 소스 컴파일

5

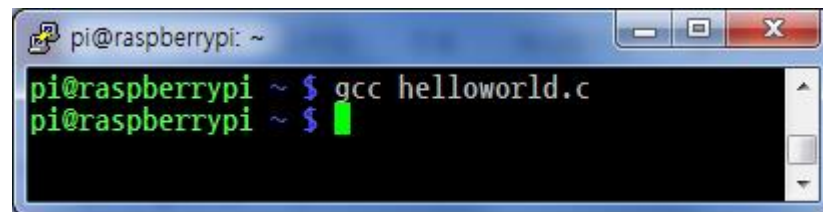
- helloworld.c 소스파일

```
#include <stdio.h>

int
main(int argc, char *argv[])
{
    printf("hello world\n");
    return 0;
}
```

- 컴파일

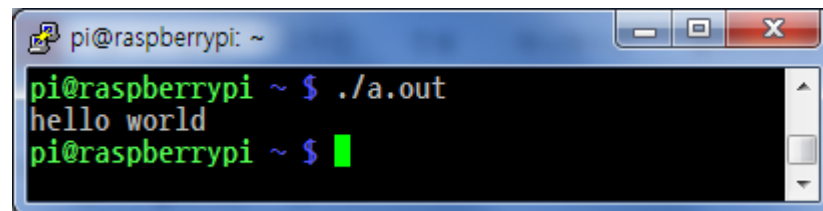
```
$ gcc helloworld.c
```

A terminal window titled 'pi@raspberrypi: ~' showing the command 'gcc helloworld.c' being executed. The prompt is 'pi@raspberrypi ~ \$' and the output is a green cursor on the next line.

```
pi@raspberrypi ~ $ gcc helloworld.c
pi@raspberrypi ~ $
```

- 실행

```
$ ./a.out
```

A terminal window titled 'pi@raspberrypi: ~' showing the command './a.out' being executed. The prompt is 'pi@raspberrypi ~ \$' and the output is 'hello world' followed by a green cursor on the next line.

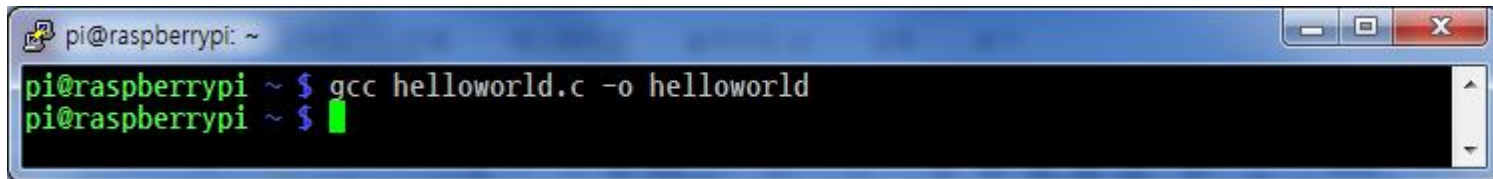
```
pi@raspberrypi ~ $ ./a.out
hello world
pi@raspberrypi ~ $
```

실습 2: hello world 소스 컴파일

6

- 컴파일(결과파일명 지정)

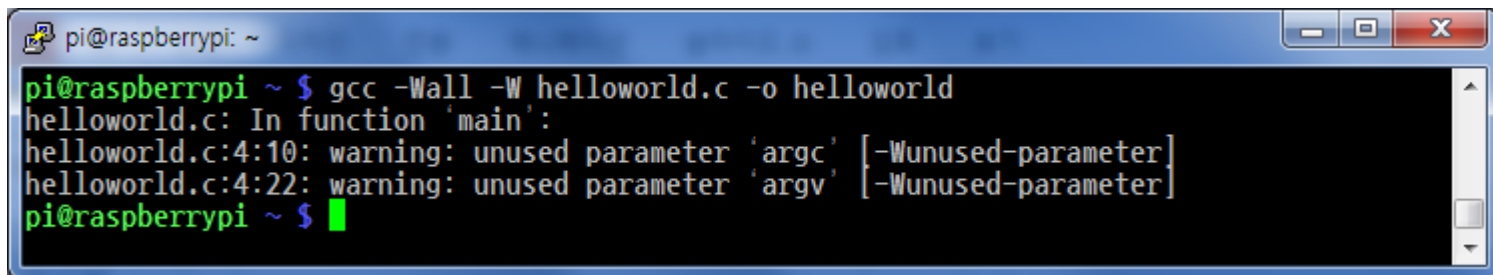
```
$ gcc helloworld.c -o helloworld
```



```
pi@raspberrypi: ~  
pi@raspberrypi ~$ gcc helloworld.c -o helloworld  
pi@raspberrypi ~$
```

- 컴파일(Warning 메시지 출력)

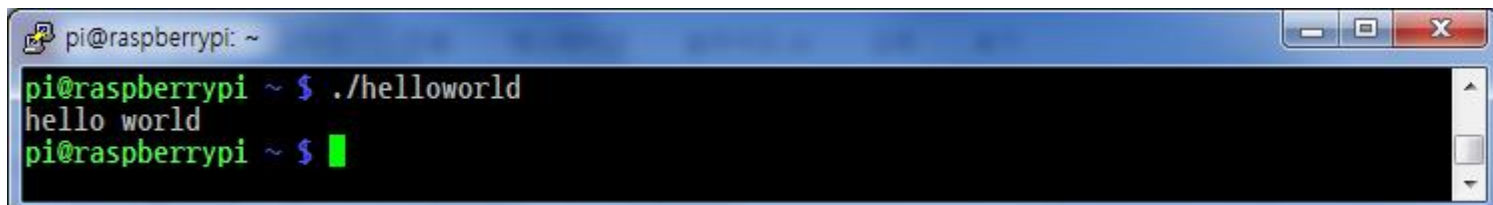
```
$ gcc -Wall -W helloworld.c -o helloworld
```




```
pi@raspberrypi: ~  
pi@raspberrypi ~$ gcc -Wall -W helloworld.c -o helloworld  
helloworld.c: In function 'main':  
helloworld.c:4:10: warning: unused parameter 'argc' [-Wunused-parameter]  
helloworld.c:4:22: warning: unused parameter 'argv' [-Wunused-parameter]  
pi@raspberrypi ~$
```

- 실행

```
$ ./helloworld
```



```
pi@raspberrypi: ~  
pi@raspberrypi ~$ ./helloworld  
hello world  
pi@raspberrypi ~$
```

 pi@raspberrypi: ~

전처리기

어셈블러

링커

미션 1 : 컴파일

9

- cla.c 소스 컴파일

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

int main(int argc, char *argv[])
{
    int i;

    for(i = 0 ; i < argc ; i++)
//    for(i = 0 ; argv[i] != NULL ; i++)
        printf("argv[%d]: %s\n", i, argv[i]);
    exit(0);
}
```

- symbol.c 소스 컴파일

```
#include <stdio.h>

#define PI 3.14
#define NUM 100
#define BUFFER_SIZE 200

int main(int argc, char *argv[])
{
    printf("%lf \n", PI);
    printf("%d \n", NUM);
    printf("%d \n", BUFFER_SIZE);

    return 0;
}
```

미션 1 : 컴파일

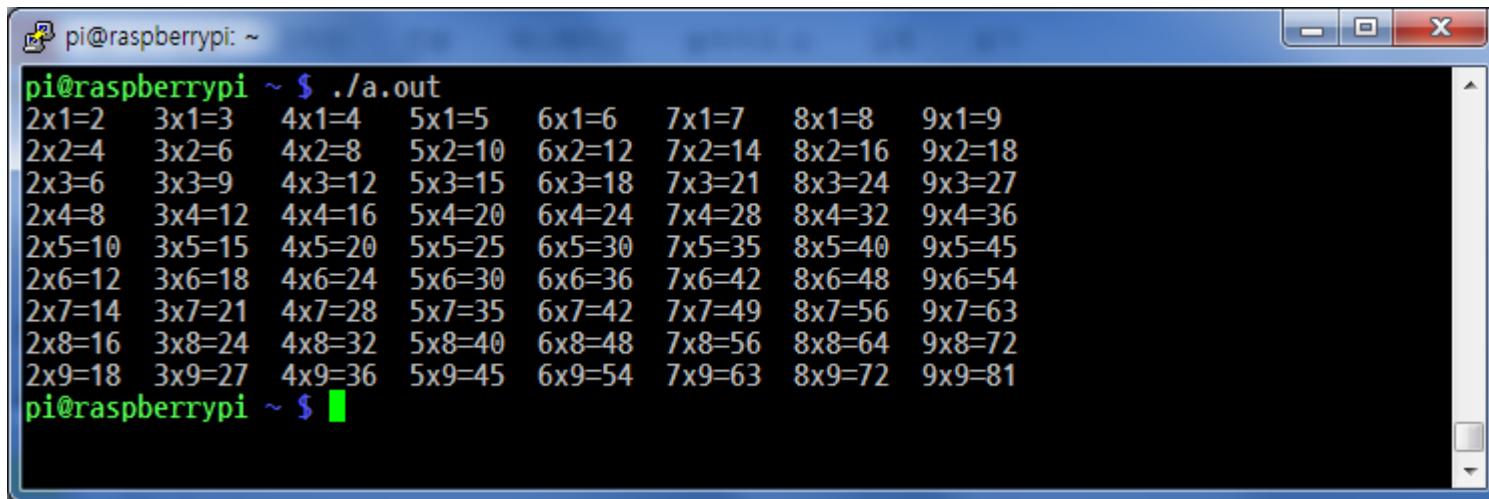
10

- symbol.c 컴파일
 - 결과파일명 : symbol
- cla.c 컴파일
 - 결과파일명 : cla

미션 2 : 컴파일

11

- 아래와 같이 구구단을 출력하는 프로그램을 작성하시오.



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ ./a.out  
2x1=2  3x1=3  4x1=4  5x1=5  6x1=6  7x1=7  8x1=8  9x1=9  
2x2=4  3x2=6  4x2=8  5x2=10 6x2=12 7x2=14 8x2=16 9x2=18  
2x3=6  3x3=9  4x3=12 5x3=15 6x3=18 7x3=21 8x3=24 9x3=27  
2x4=8  3x4=12 4x4=16 5x4=20 6x4=24 7x4=28 8x4=32 9x4=36  
2x5=10 3x5=15 4x5=20 5x5=25 6x5=30 7x5=35 8x5=40 9x5=45  
2x6=12 3x6=18 4x6=24 5x6=30 6x6=36 7x6=42 8x6=48 9x6=54  
2x7=14 3x7=21 4x7=28 5x7=35 6x7=42 7x7=49 8x7=56 9x7=63  
2x8=16 3x8=24 4x8=32 5x8=40 6x8=48 7x8=56 8x8=64 9x8=72  
2x9=18 3x9=27 4x9=36 5x9=45 6x9=54 7x9=63 8x9=72 9x9=81  
pi@raspberrypi ~ $
```

미션 3 : 컴파일

12

- 명령행 인자로 1개 이상의 정수를 입력 받아 총합과 평균을 구하는 프로그램을 작성하시오..

미션 4 : 컴파일

13

• 프로그램 내용

- main함수의 p1과 p2 포인터 변수는 target1, target2의 주소를 저장
- change 함수는 이 두 포인터 변수를 인자로 받아 서로 맞바꾸는 함수

• 이 프로그램을 다음과 같이 작성

```
#include<stdio.h>

void change(int *c1, int *c2);

int main(void)
{
    int target1 = 1111;
    int target2 = 2222;

    int *p1 = &target1;
    int *p2 = &target2;

    change(p1, p2);
    printf("%d%d\n", *p1, *p2);
    return 0;
}

void change(int *c1, int *c2 )
{
    int *temp;
    temp = c1;
    c1 = c2;
    c2 = temp;
}
```

- 이 프로그램은 11112222를 출력함
- p1 p2 포인터 변수가 서로 바뀌도록 change 함수를 올바르게 수정하시오.
 - 22221111출력