"EA0014: 유닉스프로그래밍" 실습지침서

- 실습 번호: lab-01 이름: \_\_\_\_\_\_\_\_\_서우일\_\_\_\_\_\_\_\_\_\_\_

- 실습 날짜: 2018년 03월 19일 학번: \_\_\_\_\_\_\_201311126\_\_\_\_\_\_\_\_\_

- 실습 제목: Unix, vi, 그리고 gcc

- 실습 내용:

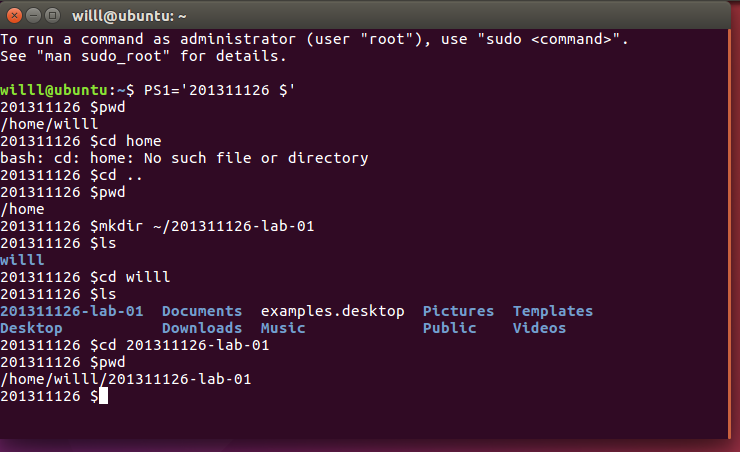
실습 환경

이번 학기 유닉스프로그래밍 실습 수업은 온라인 UNIX 터미널 사이트인 https://www.tutorialspoint.com/unix\_terminal\_online.php 에서 이루어진다. 정규 실습 시간에는 이 사이트를 통한 실습을 원칙으로 한다. 다만 실습실 밖에서 또는 개인 노트북에서 유닉스 사용을 희망하는 학생은 리눅스 가상 머신을 대여하여 개인적으로 리눅스를 사용하는 것도 가능하다

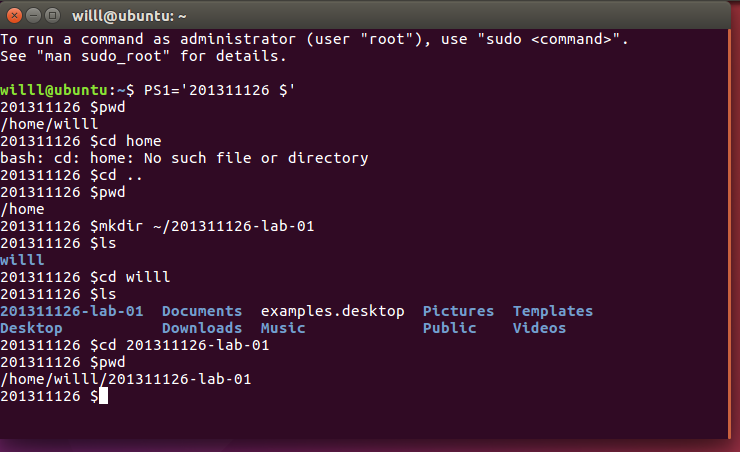
Home directory

1. 윈도우에서 웹 브라우저를 통해 https://www.tutorialspoint.com/unix\_terminal\_online.php에 접속하라. Login한 후 다음과 같이 명령을 입력하여 프람프트를 변경하라 (변경된 프람프트는 자신의 학번을 포함하게 된다. 앞으로 문제의 답으로 화면 출력 내용을 캡쳐하여 첨부하는 경우, 이 프람프트가 캡쳐 내용에 반드시 포함되도록 한다).

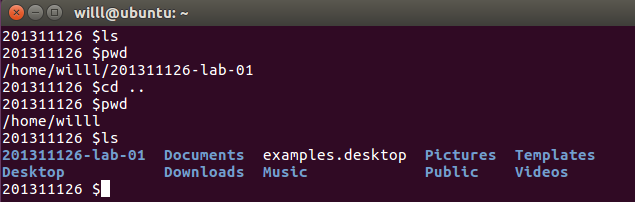
$ PS1=’\s-학번$ ‘

또한 자신의 홈 디렉토리 아래에 실습 디렉토리 ~/201611234-lab-01를 만든다(자신의 학번이 201611234인 경우). 이번 실습에서 만들어지는 모든 파일들은 이 디렉토리에 두도록 한다.

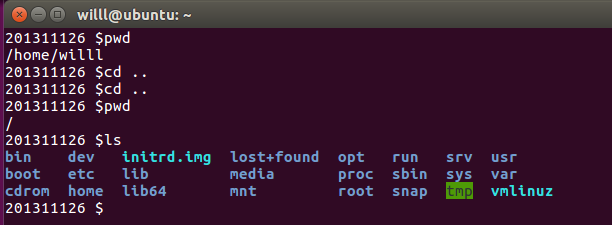
1. 현재 작업 디렉토리의 이름을 directory 이름을 프린트하는 UNIX 명령어 "pwd"를 사용하여 자기의 home directory 이름을 알아낸 후 결과를 화면 캡쳐하여 첨부하시오.



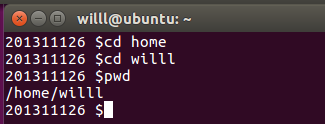
1. 현재 디렉토리에 있는 file 및 directory 이름을 프린트하는 UNIX 명령어 "ls"를 사용하여 자기의 home directory에 어떤 file 혹은 directory가 있는지 알아낸 후 결과를 화면 캡쳐하여 첨부하시오.



1. 현재 디렉토리를 다른 곳으로 옮기는 UNIX 명령어 "cd"를 사용하여 "/" directrory로 옮긴 후, 명령어 "ls"를 사용하여 어떤 file 혹은 directory가 있는지 알아보고 결과를 화면 캡쳐하여 첨부하시오.

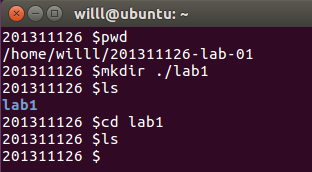


1. 그리고 "cd" 명령어를 사용하여 자기의 실습 디렉토리로 다시 돌아오라.



새 directory 만들기

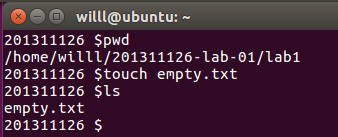
1. 새 directory를 만드는 UNIX 명령어 "mkdir"를 사용하여 새 directory “lab1”를 만들어라. 그리고 명령어 "cd"를 이용하여 directory "lab1"로 이동하시오. 여기서 명령어 "ls"를 사용하여 화면에 무엇이 출력되는지 결과를 화면 캡쳐하여 첨부하시오.



새로 만들어진 디렉토리로 들어갔으나 내부에는 .과 .. 파일이 보이지 않고 아무것도 없다고 나옵니다.

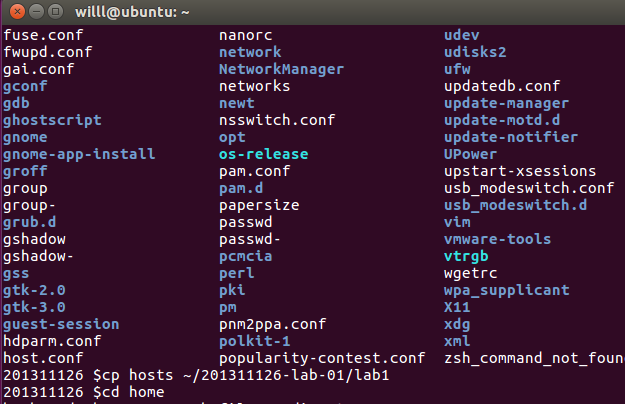
file 만들기 및 복사하기

1. directory "lab1"으로 가서 빈 text file을 하나 만드는 UNIX 명령어 "touch"를 사용하여 file 이름 "empty"인 file을 하나 만들어라. 그리고 명령어 "ls"를 사용하여 어떤 내용이 출력되는 지를 확인하고 결과를 화면 캡쳐하여 첨부하시오.



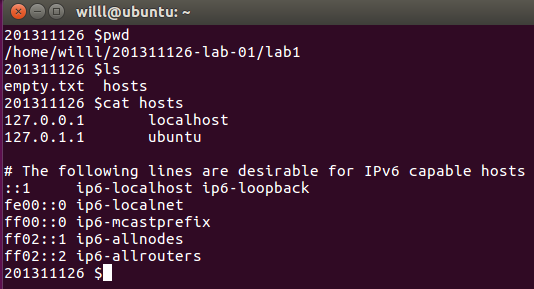
파일이 생성되었습니다.

1. directory "/etc"에는 시스템 관리를 위한 file이 많이 있다. 이들 중 file "hosts"를 file 복사하는 UNIX 명령어 "cp"를 사용하여 실습 directory에 복사하라.



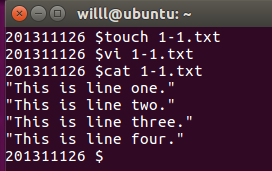
text file 내용 보기

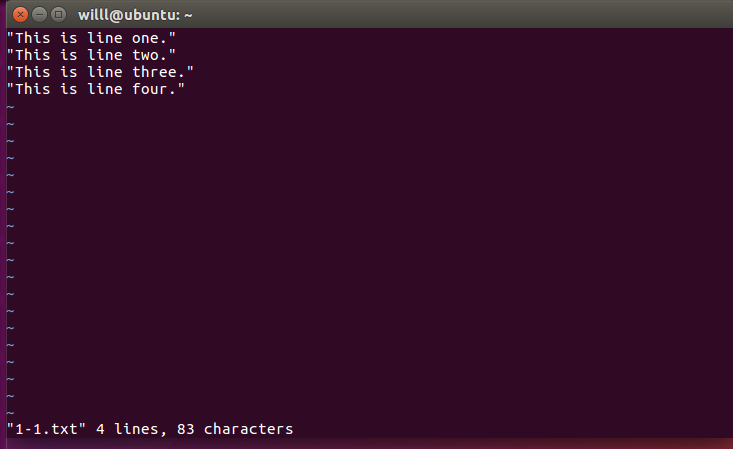
1. 앞에서 복사된 file "hosts"의 내용을 명령어 "cat"을 사용하여 읽어보고 결과를 화면 캡쳐하여 첨부하시오.



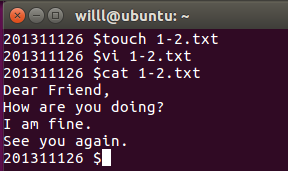
vi를 이용한 입력 및 file의 저장

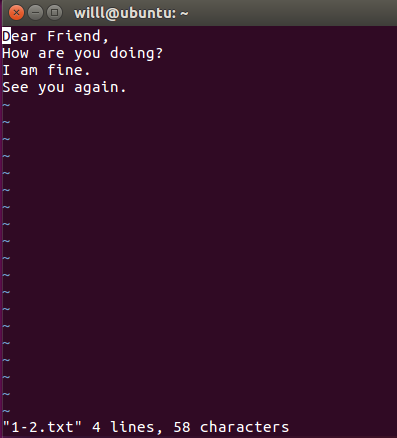
1. (file "1-1.txt") 편집기 vi를 사용하여 새로운 file "1-1.txt"를 올린 후 첫줄에는 "This is a line one.", 둘째 줄에는 "This is a line two.", 셋째 줄에는 "This is a line three.", 넷째 줄에는 "This is a line four."를 입력하고 이를 저장하라. UNIX에서 text file의 내용을 화면에 출력하는 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.





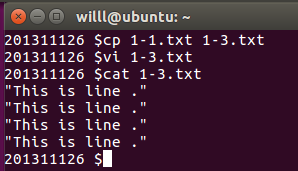
1. (file "1-2.txt") 편집기 vi를 사용하여 새로운 file "1-2.txt"를 올린 후 첫 줄에는 "Dear Friend,", 둘째 줄에는 "How are you doing?", 셋째 줄에는 "I am fine.", 넷째 줄에는 "See you again."을 입력하고 이를 저장하라. 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.



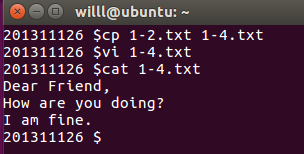


vi에서 cursor의 이동 및 삭제

1. (file "1-3.txt") 명령어 "cp"를 사용하여 앞에서 생성한 file "1-1.txt"을 file "1-3.txt"로 복사하라. file "1-3.txt"를 편집기 vi로 올린 후 첫째 줄의 "one", 둘째 줄의 "two", 셋째 줄의 "three", 넷째 줄의 "four"를 지워라. 그리고 변경된 내용을 저장하라. 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.



1. (file "1-4.txt") 명령어 "cp"를 사용하여 앞에서 생성한 file "1-2.txt"를 file "1-4.txt"로 복사하라. file "1-4.txt"를 편집기 vi로 올린 후 넷째 줄의 내용 "See you again."을 모두 지워라. 그리고 변경된 내용을 저장하라. 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.



간단한 프로그램 편집

1. (file "simple.c") 편집기 vi를 사용하여 다음 내용을 가지는 간단한 C 프로그램 file "simple.c"를 생성하라. 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.

#include <stdio.h>

#include <stdlib.h>

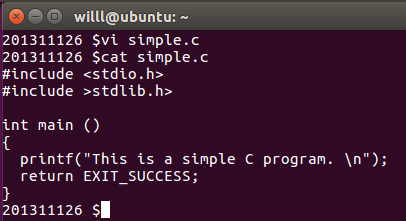
int main()

{

printf("This is a simple C program.\n");

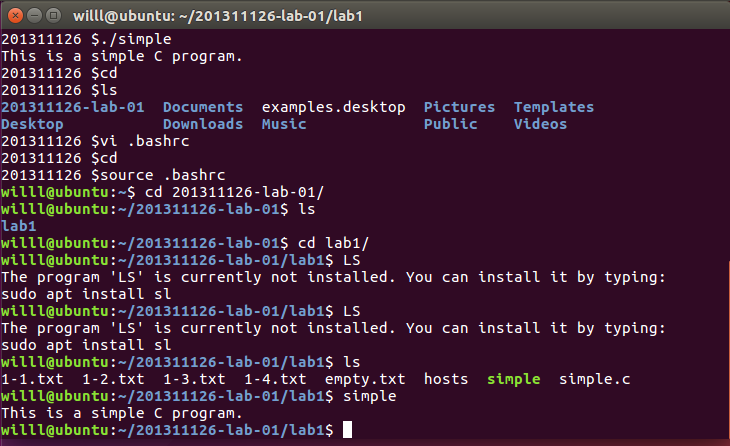
return EXIT\_SUCCESS;

}



1개 C 프로그램 file compile 방법

1. (file "simple.c") 앞에서 작성한 C 프로그램 "simple.c"를 명령어 "gcc simple.c -o simple"를 사용하여 compile한 후 수행 file "simple"을 execute하여 그 결과를 확인하라. compile 시 error message가 나오면 이를 수정하여 다시 compile 한 후 execute시켜라. 이 프로그램의 execute 시 화면에 어떤 메시지가 출력되는지 결과를 화면 캡쳐하여 첨부하시오.



저의 경우 우분투 가상머신으로 실행했던 경우인데, 시스템 내에 다른 simple 파일과 겹쳐서 Shell이 정확히 인지하지 못한 경우가 있어서 교수님이 도와주셨습니다. 같은 이름의 파일이 존재할 시에 경로 설정을 해주지 않으면 충돌이 있을 수 있다는 것을 이 경우로 인해 알게 되었습니다.

1개 C 프로그램 file preprocess/compile/link 방법

1. (file "step.c") 편집기 vi 사용하여 다음 내용의 간단한 C program file "step.c"를 생성하고, 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.

#define PRINCIPAL 5000.00

#define INTRATE 0.06

#define EXIT\_SUCCESS 0

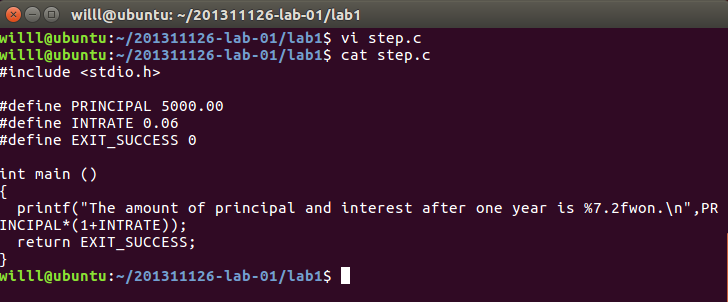
int main()

{

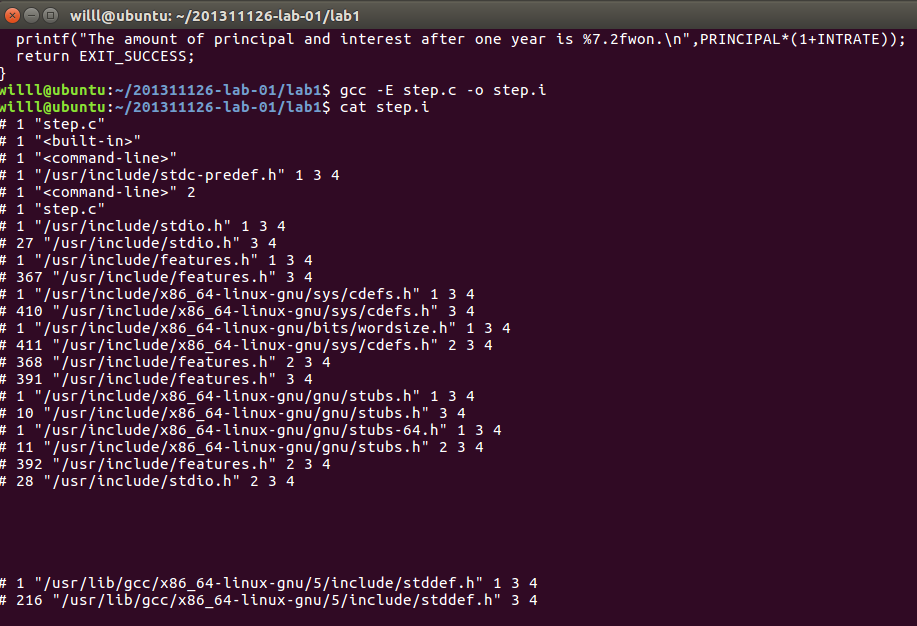
printf("The amount of principal and interest after one year is %7.2f won.\n", PRINCIPAL\*(1+INTRATE));

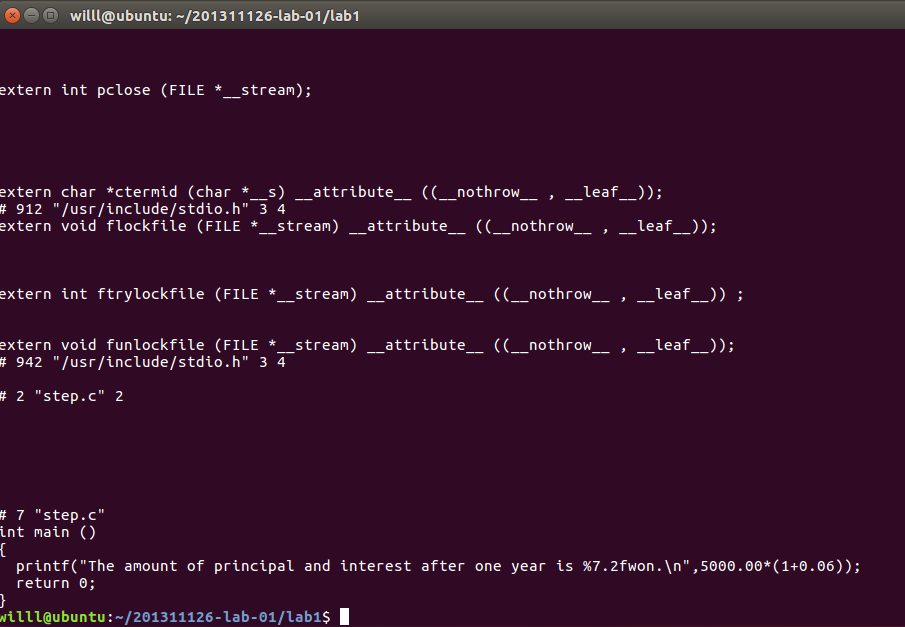
return EXIT\_SUCCESS;

}



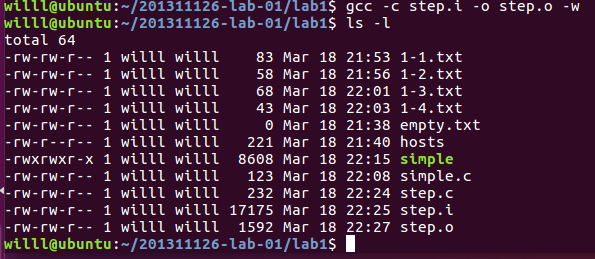
1. (file "step.i") C program file step.c를 preprocess하려고 한다. 명령어 "gcc -E step.c -o step.i"를 사용하여 file "step.c"를 preprocess하여 file "step.i"를 생성하고, 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.



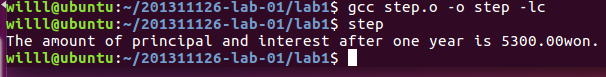


배운 내용에 의하면, step.i파일은 파일내의 내용을 전처리하여 자체적으로 해석한 내용이라고 생각됩니다. 너무 길어서 스크린샷은 맨 상단과 최하단만 찍었습니다.

1. (file "step.o") 앞의 preprocessor에 의하여 생성된 file "step.i"를 compile 하려고 한다. 명령어 "gcc -c step.i -o step.o -w"를 사용하여 file "step.i"를 compile하여 object file "step.o"를 생성하라. (주의: file "step.o"는 기계어 프로그램 file이기 때문에 명령어 "cat"을 사용하여 그 내용을 볼 수는 없다.) 명령어 "ls -l"을 사용하여 "step.o"의 크기를 확인하고 결과를 화면 캡쳐하여 첨부하시오.



1. (file "step") 앞의 compiler에 의하여 생성된 file "step.o"와 C library를 link하려고 한다. 명령어 "gcc step.o -o step -lc"를 사용하여 executable file "step"을 생성하라. 생성된 executable file을 수행시키면 화면에 어떤 내용이 출력되는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.



여러개 C 프로그램 file compile 방법

1. (file "file1.c") 편집기 vi 사용하여 다음 내용의 간단한 C program file “file1.c” 를 생성하고, 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.

#include <stdio.h>

#include <stdlib.h>

int hello();

int main()

{

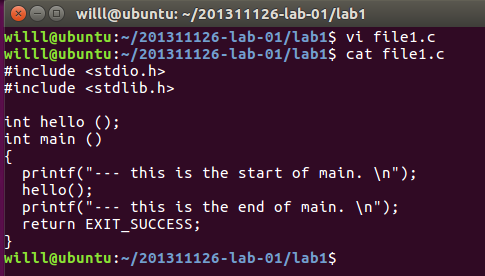
printf("--- This is the start of main.\n");

hello();

printf("--- This is the end of main.\n");

return EXIT\_SUCCESS;

}



1. (file "file2.c") 편집기 vi 사용하여 다음 내용의 간단한 C program file “file2.c” 를 생성하고, 명령어 "cat"을 사용하여 이 file이 잘 만들어 졌는지 확인하고 결과를 화면 캡쳐하여 첨부하시오.

#include <stdio.h>

#include <stdlib.h>

int hello()

{

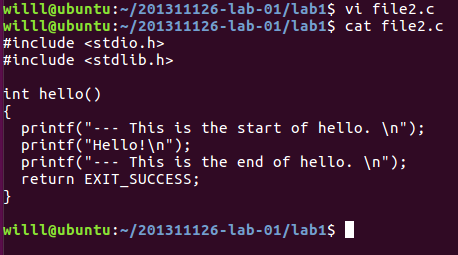
printf("--- This is the start of hello.\n");

printf("Hello!\n");

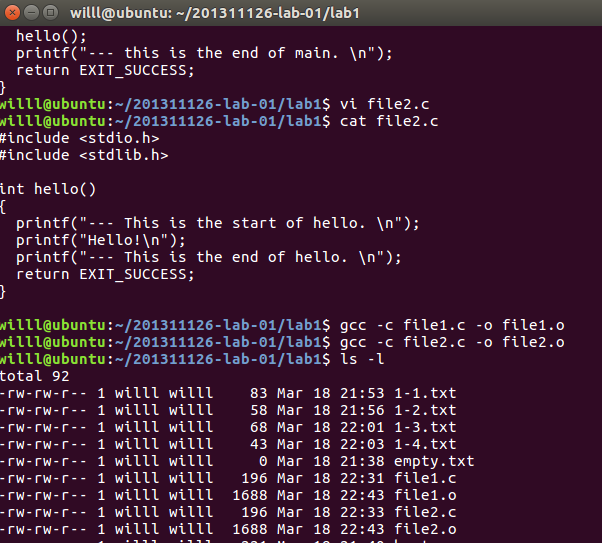
printf("--- This is the end of hello.\n");

return EXIT\_SUCCESS;

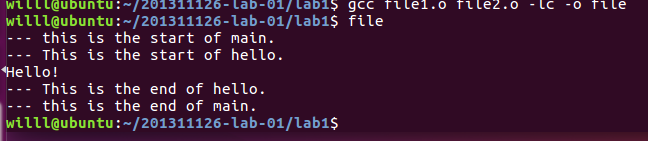
}



1. (file "file1.o", "file2.o") 명령어 "gcc -c file1.c -o file1.o"을 사용하여 C 프로그램 file "file1.c"를 compile하여 object file "file1.o"를 생성하라. 명령어 "gcc -c file2.c -o file2.o"을 사용하여 C 프로그램 file "file2.c"를 compile하여 object file "file2.o"를 생성하라. 명령어 "ls -l"을 사용하여 "file1.o" 및 "file2.o"의 크기를 확인하고 결과를 화면 캡쳐하여 첨부하시오.



1. (file "file") 명령어 "gcc file1.o file2.o -lc -o file"을 사용하여 object file "file1.o", "file2.o", 및 C library를 link하여 executable file "file"을 만들어라. executable file "file"을 수행시켜 그 결과를 화면 캡쳐하여 첨부하시오.



1. (실습 결과물 제출) 실습 결과는 e-campus를 통해 다음 두 가지 파일을 제출한다. 실습 시간 중에 완성하지 못한 경우, 나머지 부분은 나중에 완성하여 제출할 수 있다. 제출 마감 시간은 실습한 날로부터 3일 후 자정까지이다 (예를 들어 월요일에 실습을 했다면, 목요일 밤 12시까지 제출하면 된다).
2. 답안이 작성된 본 실습지침서 파일; 단, 파일 이름은 반드시 학번-lab-01-unix-vi-gcc.docx 형태이어야 한다. 예를 들어 자신의 학번이 201611234인 경우, 실습지침서 파일의 이름은 201611234-lab-01-unix-vi-gcc.docx로 하여 e-campus에 업로드한다.

실습 중 생성된 모든 파일 (tar 파일로 묶어서 제출); 이번 실습 과정에서 생성된 모든 파일은 자신의 홈 디렉토리 아래 ~/201611234-lab-01 디렉토리에 저장되어 있다.  이를 자신의 홈디렉토리에서 아래 명령을 실행하여 하나의 tar 파일로 묶고,  생성된 tar 파일을 자신의 PC로 다운로드한 후,  이를 e-campus에 제출한다.

$ tar cvf 201611234-lab-01.tar 201611234-lab-01/

실습 디렉토리 이름

생성할 tar 파일 이름

참고: gcc 명령어 option의 뜻

----------------------------

$ gcc prog.c -o prog ("prog.c"를 preprocess/compile/link하여 executable file "prog"를 만들어라.)

$ gcc -E prog.c -o prog.i ("prog.c"를 preprocess하여 "prog.i"를 만들어라.)

$ gcc -c prog.i -o prog.o ("prog.i"를 compile하여 object file "prog.o"를 만들어라.)

$ gcc prog1.o prog2.o -lc -o prog (object file "prog1.o", "prog2.o", 및 C Library를 link하여 executable file "prog"를 만들어라.)

끝.