

프로그래밍 연습

실습#2

Linux

I. Linux란?

II. Linux 파일시스템

III. Linux 기본 명령어

I. Linux란?

운영체제

- 컴퓨터의 하드웨어 자원들을 제어
- 프로그램이 실행될 환경을 제공

Linux를 배우는 이유

- 무료 & 오픈소스
 - 운영체제 자체를 임의로 수정 가능
 - 전세계적으로 많은 개발 및 연구 그룹 형성하고 정보를 공유

Linux의 특징

- C언어를 기반으로 제작
- 기본적으로 명령어들로 조작

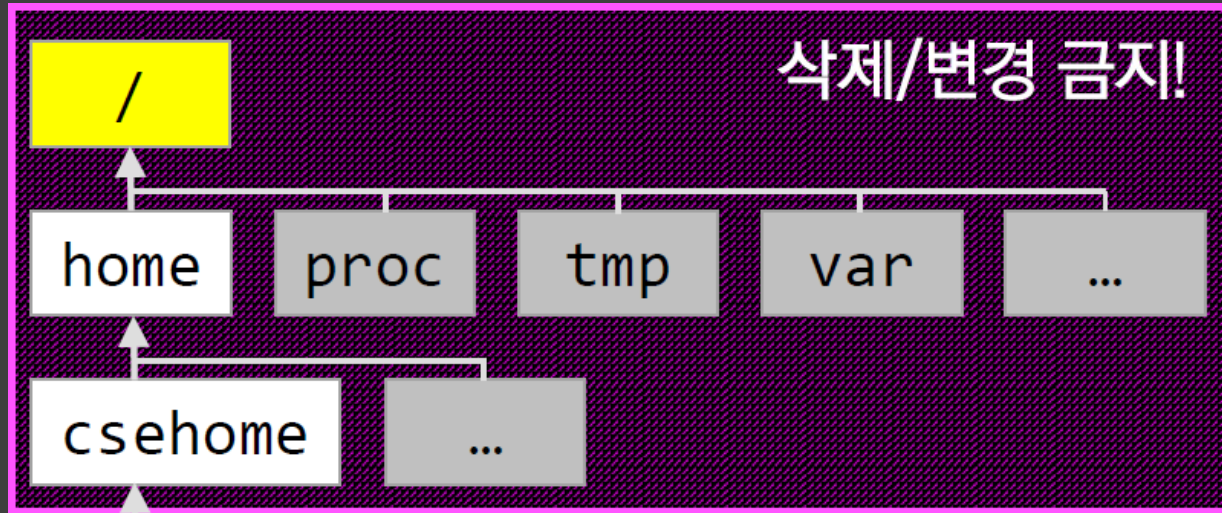
II. Linux 파일 시스템

파일 시스템이란?

- 파일을 저장하고 구조화하는 체계
- 파일을 쉽게 찾고 접근할 수 있게 함

파일 시스템 구조

루트 디렉터리



홈 디렉터리 (계정명) : /home/csehome/(계정명)

II. Linux 파일 시스템

루트 디렉터리(/)

- 파일시스템에서 최상위 디렉터리
- 모든 파일 및 디렉터리는 루트로부터 시작
- '/'로 표현

홈 디렉터리(/home/csehome/계정명, ~)

- 각 사용자에게 할당된 디렉터리
- '~'로 표현

II. Linux 파일 시스템

작업 디렉터리 (.)

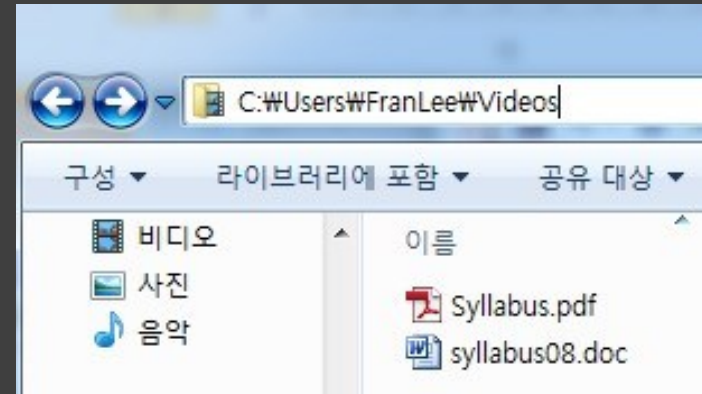
- 파일시스템 상에서 현재 위치
- '.'로 표현

부모 디렉터리

- 작업 디렉터리의 바로 상위 디렉터리
- '..'로 표현

Windows에서의 예:

C:\Users\CSESNU\Videos



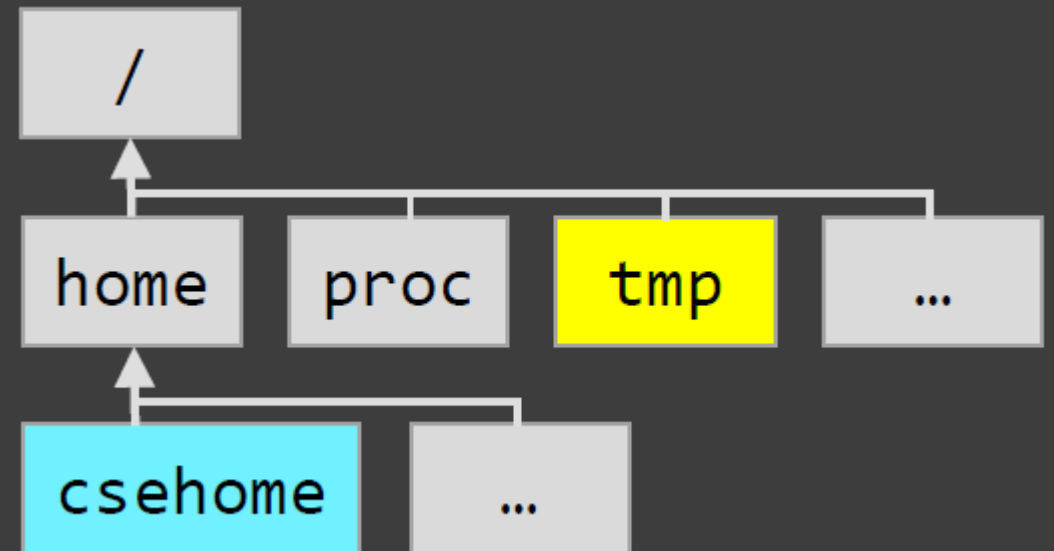
II. Linux 파일 시스템

경로

- 파일시스템 내에서 특정 위치를 표현하는 것
- 절대경로
 - 루트 디렉터리부터 시작하는 경로
- 상대경로
 - 현재 디렉터리를 기준으로 표현되는 경로

예: 작업 디렉터리가 csehome 일 때

- csehome의 경로
 - 절대경로 : `/home/csehome`
 - 상대경로 : `./`
- tmp의 경로
 - 절대경로 : `/tmp`
 - 상대경로 : `../tmp`



I/O Redirection

1. 입출력 재지정이란?

1. 입출력 재지정이란?

기본 입출력

- 표준 출력 : 모니터, 표준 입력 : 키보드


입출력 재지정

- 모니터가 아닌 **파일에 출력 결과를 저장**하거나,
키보드가 아닌 **파일을 입력**으로 설정할 수 있음
- 입력 재지정 : <
- 출력 재지정 : > 혹은 >>
 - 출력 파일이 존재하지 않으면 새로 파일이 생성됨
 - 출력 파일이 이미 존재한다면
 - > : 기존 내용은 **모두 지우고 새로** 작성
 - >> : 기존 내용에 **이어서** 출력결과 작성


I. 입출력 재지정이란?

입출력 재지정

- 예 : $a < b$ b파일의 내용을 a의 입력으로 설정



$a > b$ 프로그램 a의 출력 결과를 b에 저장



$a.out < input_file >> output_file$

1. input_file 내용을 a.out의 입력으로 설정
2. a.out의 출력 결과를 output_file에 저장

실습환경

`martini.snucse.org`

- 자신의 `martini` 계정/임시계정으로 실습컴퓨터 로그인
- window키>왼쪽 'terminal' 켜기
- `ssh {id}@martini.snucse.org`
ex) `ssh PPmcc127@martini.snucse.org`

Linux 기본 명령어

1) ls

- 파일 목록을 출력하는 명령어
- 사용법 : ls [옵션][경로]
 - 주요 옵션
 - a : 모든 파일보기
 - r : 하위 디렉터리의 파일까지 보기

```
ppmcc127@martini:~$ ls  
hello_world.c  
ppmcc127@martini:~$
```



2) mkdir

- 디렉터리를 생성하는 명령어
- 사용법 : mkdir [이름]

```
ppmcc127@martini:~$ mkdir class2  
ppmcc127@martini:~$ ls  
class2  hello_world.c  
ppmcc127@martini:~$
```



Linux 기본 명령어

3) cd

- 디렉터리로 이동하는 명령어
- 사용법 : cd [directory]
- ./ 현재 디렉토리
- ../ 부모 디렉토리
- ~/ 홈 디렉토리

```
ppmcc127@martini:~$ cd class2  
ppmcc127@martini:~/class2$
```

```
ppmcc127@martini:~/class2$ cd ..  
ppmcc127@martini:~$
```

4) rmdir

- 빈 디렉터를 삭제하는 명령어
 - 파일이 단 하나라도 포함된 디렉터를 삭제할 때에는 rm 명령어를 사용
- 사용법 : rmdir [이름]

vim 편집기

vim 파일명.확장자

- 새로운 파일 만들기 / 기존 파일 열기
- 프로그램 언어에 따라 확장자 설정(e.g.test.c, test.cpp)

vim 편집기에서의 모드

- Normal mode
 - Vim의 초기 모드
- Insert mode
 - 편집모드
 - 'i'를 누르면 normal mode에서 insert mode로 전환
 - 'esc'를 누르면 insert mode에서 normal mode로 전환

vim 편집기

Normal mode에서의 명령어

- **:w** 문서 저장
- **:q** vim 종료
- **:wq** 문서 저장 후 vim 종료
- **:q!** 문서 강제로 저장하지 않고 vim 종료
- **u** undo
- **dd** 해당 line 잘라내기
- **yy** 해당 line 복사하기
- **p** 복사한 line 붙여넣기

실습

실습1

1. vim ex1.c 입력합니다.
2. i를 눌러 입력모드로 변경한 후 다음 코드를 작성해 보세요.

```
#include <stdio.h>
int main(void)
{
    int age = 0;

    printf("Enter your age\n");
    scanf("%d", &age);
    printf("I'm %d years old\n", age);

    return 0;
}
```

실습1

➤ 코드분석

```
int age = 0;
```

- 정수형 변수 `age`를 선언하고, 초기값은 `0`으로 설정

```
printf("Enter your age\n");
```

- `printf` : 표준출력(모니터)에 문자들을 출력하는 함수
 - ✓ `printf("")` : 큰 따옴표 안에 내용을 출력하겠다.
 - ✓ `printf("Enter your age")` :
Enter your age 라는 문장을 출력하겠다.

실습1

➤ 코드분석

`scanf("%d", &age);`



- scanf : 문자를 표준입력(키보드)로부터 읽어 들이는 함수
 - ✓ scanf("") : 큰 따옴표 안에 내용을 읽어 들이겠다.
 - ✓ scanf("%d") : 정수 값을 읽어 들이겠다.
 - ✓ %d : 정수형 값을 읽어들이는 키워드
 - ✓ scanf("%d", &age) :
읽어 들인 정수 값을 변수 age에 저장하겠다.
 - ✓ &를 반드시 붙일 것!

실습1

➤ 코드분석

`printf("I'm %d years old\n", age);`



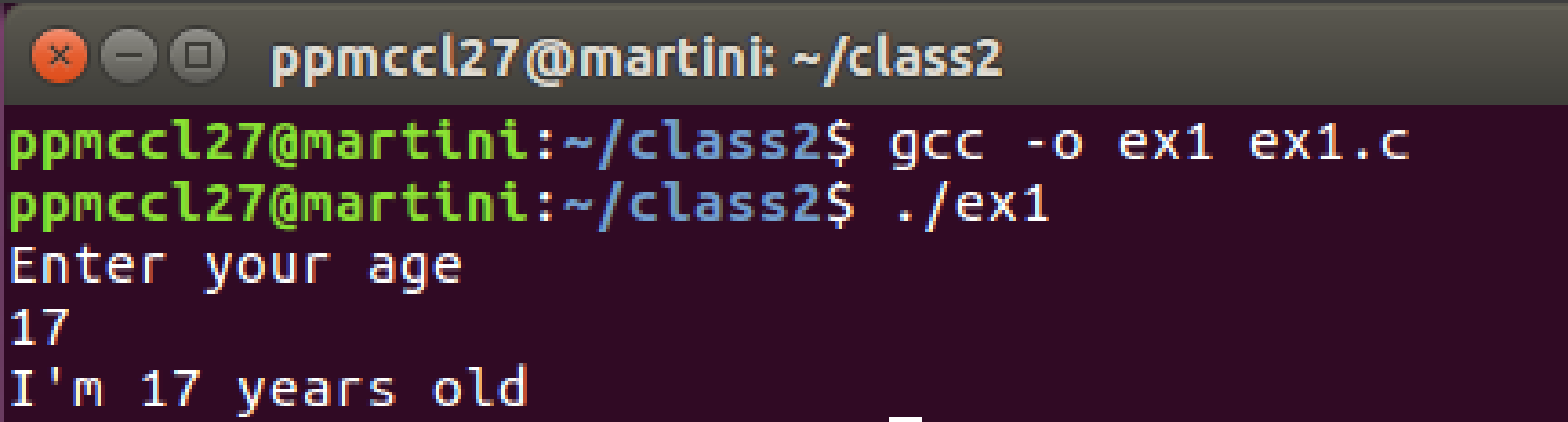
- `printf` : 표준출력(모니터)에 문자들을 출력하는 함수
 - ✓ `printf("")` : 큰 따옴표 안에 내용을 출력하겠다.
 - ✓ `printf("%d")` : 정수 값을 출력하겠다.
 - `%d` : 해당 자리에 정수형 값을 출력한다는 키워드
 - ✓ `printf("%d", age)` :
정수형 변수 `age`에 저장되어 있는 값을 출력하겠다.

`%d` 자리에는 반드시 정수형(int)값만을 읽거나 출력할 수 있다.

실습1

➤ 실행

3. Esc를 눌러 명령모드로 변경하세요.
4. :wq를 입력하고 enter키를 누르세요. (저장 후 vim 종료)
5. 컴파일을 실행: **gcc -o ex1 ex1.c**
 -o : 실행파일명 설정 (기본 : a.out)
6. ./ex1 을 입력해 작성한 프로그램을 실행합니다.
7. 본인 나이를 입력해보세요 (**반드시 숫자로!**)



```
ppmccl27@martini: ~/class2
5 ppmccl27@martini:~/class2$ gcc -o ex1 ex1.c
6 ppmccl27@martini:~/class2$ ./ex1
Enter your age
7 17
I'm 17 years old
```

실습2

원의 둘레(**circumference**)와 넓이(**area**)를 계산하는 프로그램을 작성한다.
지름 값을 입력 받아 해당하는 원의 둘레와 넓이를 구한다. 원주율은 3.14를 사용한다.

출력은 다음과 같이 되도록 한다.

```
1 ppmccl27@martini:~$ ./ex2
지름(diameter)의 값을 입력하세요.
12
둘레(circumference): 37.680000
넓이(area): 113.040001
2 ppmccl27@martini:~$ ./ex2
지름(diameter)의 값을 입력하세요.
100
둘레(circumference): 314.000000
넓이(area): 7850.000488
ppmccl27@martini:~$
```

과제

과제1

1차, 2차, 3차 시험 점수를 입력 받아 평균을 구한 후, PASS/FAIL 을 판별하는 프로그램을 작성한다. 평균이 80점 이상일 경우 PASS를 출력하고 이하일 경우 FAIL을 출력한다.

출력은 다음과 같이 되도록 한다.

```
① ppmcc127@martini:~$ ./hw1
1차, 2차, 3차 점수를 입력하세요.
75 90 95
PASS

② ppmcc127@martini:~$ ./hw1
1차, 2차, 3차 점수를 입력하세요.
90 54 54
FAIL

ppmcc127@martini:~$
```


과제2

홀수와 짝수의 개수를 구하는 프로그램을 작성한다. N의 값을 입력 받아 1부터 N까지의 자연수 중 홀수와 짝수의 개수를 출력한다.

출력은 다음과 같이 되도록 한다.

```
① ppmcc127@martini:~$ ./hw2
N의 값을 입력하세요.
10000
홀수 : 5000 개 , 짝수 : 5000 개

② ppmcc127@martini:~$ ./hw2
N의 값을 입력하세요.
12345
홀수 : 6173 개 , 짝수 : 6172 개
ppmcc127@martini:~$
```

※추가 자료

vim 편집기 설정

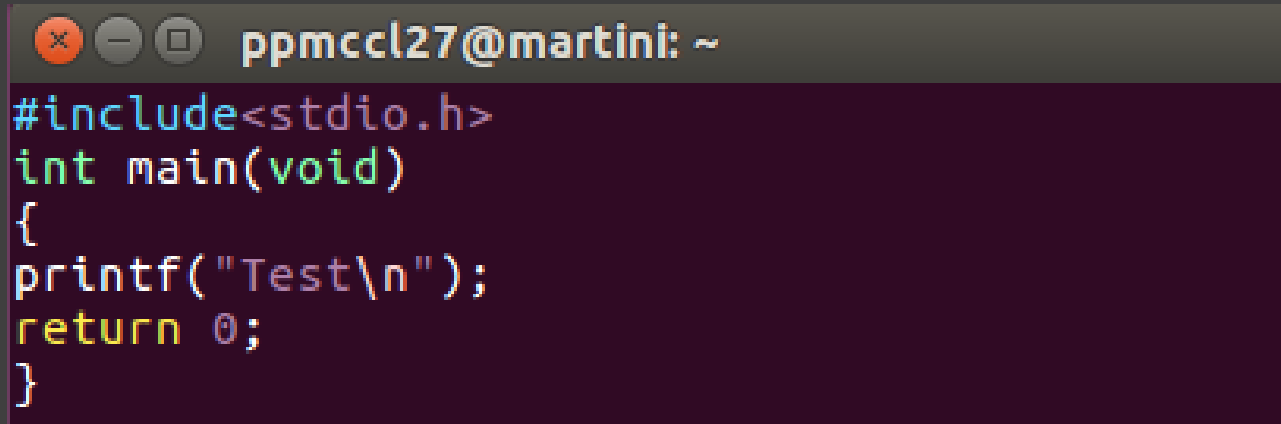
Vim 편집기 설정 변경

대표 설정

- `set number` “line 표시
- `set autoindent` “자동 들여쓰기
- `set smartindent`
- `set cindent` “c언어 자동 들여쓰기
- `set shiftwidth=4` “자동 들여쓰기 너비 설정
- `set tabstop=4`
- `set ruler` “현재 커서 위치 표시

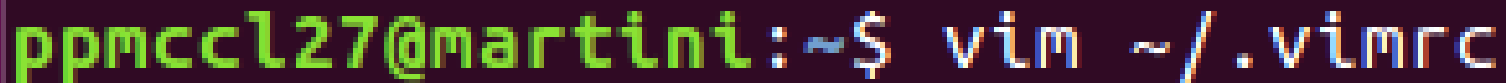
Vim 편집기 설정 변경

1. 편집기 설정 전 hello_world.c 화면



```
ppmccl27@martini: ~  
#include<stdio.h>  
int main(void)  
{  
printf("Test\\n");  
return 0;  
}
```

2. 편집기 설정을 위해 vim ~/.vimrc 입력



```
ppmccl27@martini:~$ vim ~/.vimrc
```

Vim 편집기 설정 변경

3. 아래와 같이 설정 내용 입력



```
ppmcc127@martini: ~
```

```
set number
set autoindent
set smartindent
set cindent
set shiftwidth=4
set tabstop=4
set ruler
```

PM

4. :wq

ppmccl27@martini: ~

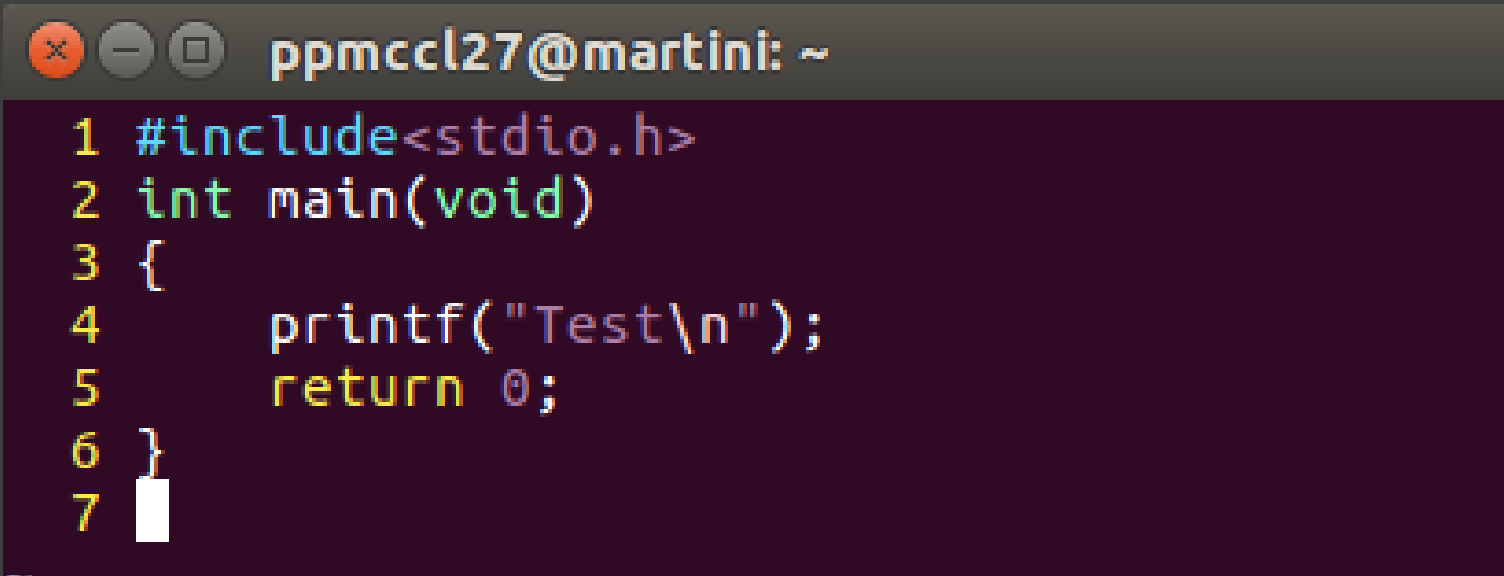
```
set number
set autoindent
set smartindent
set cindent
set shiftwidth=4
set tabstop=4
set ruler
```

~~~~~

: WQ

## Vim 편집기 설정 변경

5. 다시 hello\_world.c 로 들어가보면 편집기가 변경된 걸 확인할 수 있음

A terminal window with a dark background and light-colored text. The window title bar shows standard Linux window controls (close, maximize, and a button labeled 'ppmccl27@martini: ~'). The terminal content shows a C program being edited in Vim. The code is as follows:

```
1 #include<stdio.h>
2 int main(void)
3 {
4     printf("Test\n");
5     return 0;
6 }
7
```

The cursor is positioned at the end of line 7.