



1번째 숙제

임베디드스쿨2기

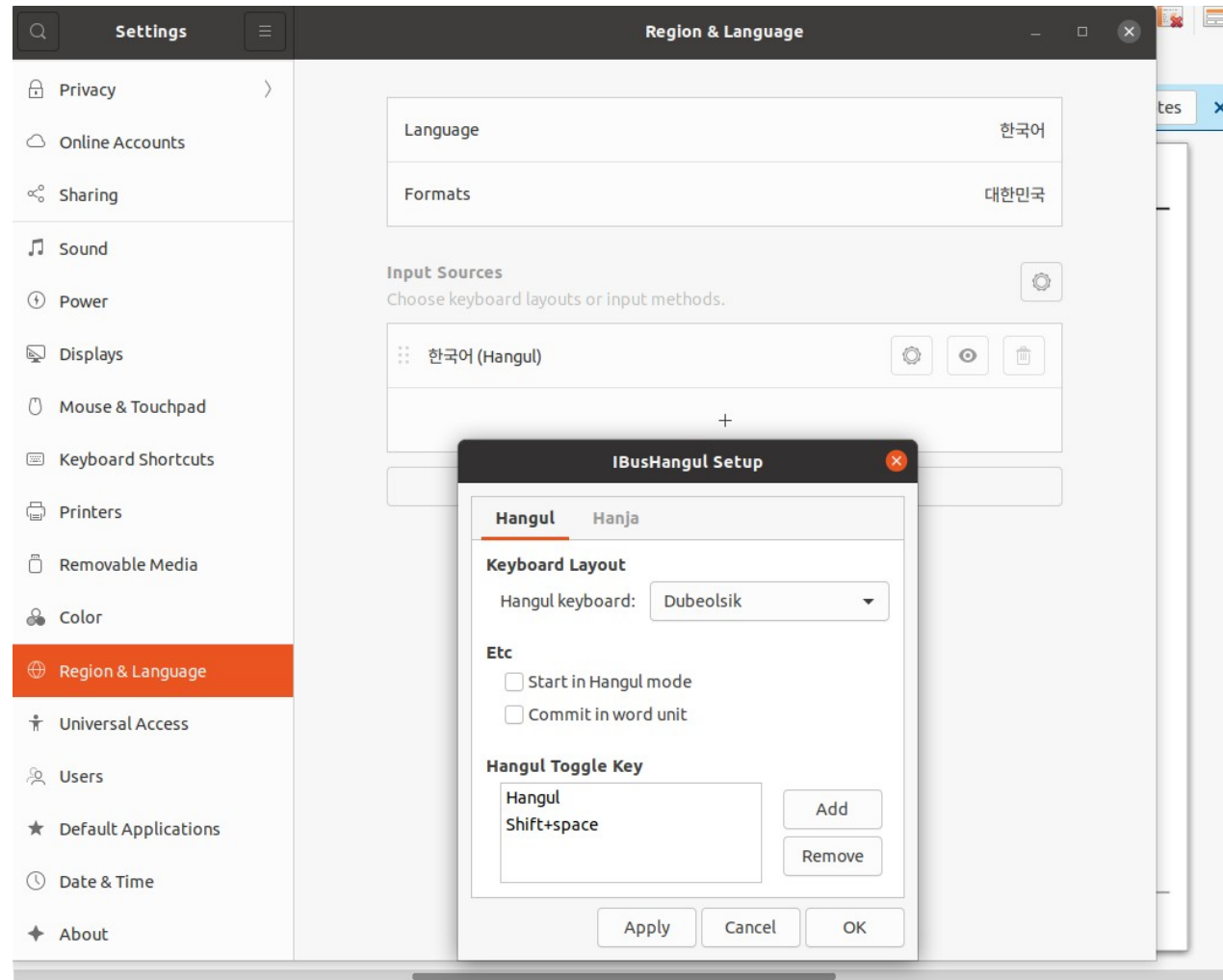
Lv1과정

2021. 03. 18

박태인

1. 초기 설정

1) 우분투 설치 후 한글 설정 방법.



Setting 을 검색후 지역&언어 탭을 선택하고 input sources의 설정 탭에서 한글 입력 방법을 확인한다.
현재 한영키와 shift+space로 되어 있기는 하나 실제로는 shift+space키를 해야 먹히는 듯 하다.

1. 초기 설정

2) 기본 프로그램 설정

// 프로그램 설치 패키지 업데이트

Sudo apt-get update

//편집기 설치

Sudo apt-get install vim

//git 설치

Sudo apt-get install git

//gcc 컴파일러 설치

Sudo apt install gcc

```
taein@taein-Lenovo-ideapad-700-15ISK:/home$ sudo apt-get update
받기:1 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
받기:2 http://kr.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
받기:3 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [204 kB]
받기:4 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
받기:5 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
받기:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [547 kB]
```

```
taein@taein-Lenovo-ideapad-700-15ISK:/home$ sudo apt-get install vim
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
The following additional packages will be installed:
  vim-runtime
제안하는 패키지:
  ctags vim-doc vim-scripts
```

```
taein@taein-Lenovo-ideapad-700-15ISK:/home$ sudo apt-get install git
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
The following additional packages will be installed:
  git-man liberror-perl
제안하는 패키지:
```

주의) 인터넷이 연결 된
상태에서 진행 하여야 한다.

1. 초기 설정

3) 자신의 저장소를 설정. (fork 계정)

1. <https://github.com/EmbeddedSchoolReop/Lv01-02> 로 이동합니다.
↳ (fork 해놓은 github 주소)
2. `cd ~` (~ 로 이동)
3. `mkdir proj` (proj 파일 생성)
4. `cd proj` (proj 로 이동)
5. `mkdir es02` (es02 파일 생성)
6. `cd es02` (es02 로 이동)
7. `git clone "자신의 fork 주소 복사"` (git 주소 복사)
8. `ls` (폴더 상태 확인)

```
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
taein@taein-Lenovo-ideapad-700-15ISK:/$ cd ~
taein@taein-Lenovo-ideapad-700-15ISK:~$ mkdir proj
taein@taein-Lenovo-ideapad-700-15ISK:~$ cd proj
taein@taein-Lenovo-ideapad-700-15ISK:~/proj$ mkdir es02
taein@taein-Lenovo-ideapad-700-15ISK:~/proj$ cd es02
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02$ git clone https://github.com/parktaein/Lv01-02.git
Cloning into 'Lv01-02'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 29 (delta 11), reused 15 (delta 4), pack-reused 0
Unpacking objects: 100% (29/29), 6.31 KiB | 1.26 MiB/s, done.
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02$ ls
Lv01-02
```

1. 초기 설정

4) Git 활용 설정

1. 현재 앞 전의 작업을 통해 저장소를 가지고 있을 것이다.
2. `git config --global user.name "깃 사용자 명"`
3. `git config --global user.email "깃 가입 메일"`
4. `git config --list` 를 통해 정상적으로 처리가 되었는지 확인 합니다.
5. 이름에 해당하는 디렉토리로 이동해서 작업(숙제) 진행.
6. `git status` 를 입력하면 현재 상태를 알 수 있음.
7. 빨간색으로 나오는 정보들을 `git add` 형태로 추가하면 됩니다.
8. 이후 `git commit` 형식으로 제출
9. `commit` 이후에 `git push origin main` 으로 각자의 사본 저장소에 저장
10. 개인 사본 저장소에서 Pull Request를 눌러서 PR 요청 넣으면 완료
11. 이후 관리자가 승인하면 원본 저장소에 내용 갱신 됨.

```
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02$ cd Lv01-02
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02$ ls
ChungjaeLee HyochangKim HyunhoCha LICENSE README.md TaeinPark init.md
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02$ cd TaeinPark
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark$ ls
prepare.txt  text.txt
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark$ mkdir homework
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark$ ls
homework  prepare.txt  text.txt
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark$ cd homework
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework$ ls
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework$ mkdir -p c/01/
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework$ ls
c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework$ cd c/01
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
```


1. 초기 설정

4) Git 활용 설정

```
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://github.com/parktaein/Lv01-02.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ git config --global user.name
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ git config --list
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://github.com/parktaein/Lv01-02.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ git config --global user.name "TaeinPark"
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ git config --list
user.name=TaeinPark
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://github.com/parktaein/Lv01-02.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
```

2. 리눅스 명령어(1)

- ls → list의 약자로 현재 디렉토리의 리스트를 보여 줍니다.
 - 사용법 1) **ls**
 - 사용법 2) **ls** 디렉토리 명
 - 사용법 3) **ls -R** (현재 위치부터 하위의 **모든 정보**를 보여줌)
 - 사용법 4) **ls -a**
리눅스에서 숨김 파일은 맨 앞에 '.'이 온다.
숨김파일을 포함해서 리스트를 보여 준다.
- clear → 화면이 아래쪽으로 내려가서 맨위로 올리고 싶은 경우 혹은 화면이 지저분해서 정리하고 싶은 경우
- cp → 복사 명령어
 - 사용법 1) **cp** 복사대상 사본명
 - 사용법 2) **cp -r** 복사대상(**디렉토리**) 사본명
- mkdir → make directory의 약자로 디렉토리를 생성합니다.
 - 사용법 1) **mkdir** 디렉토리명
 - 사용법 2) **mkdir -p** 디렉토리명1/디렉토리명2/디렉토리명3/
이와 같이 만들면 디렉토리1 내부에 디렉토리2가 만들어지고 디렉토리2 내부에 디렉토리3 생성.
- cd → change directory의 약자로 디렉토리를 이동하는 목적으로 사용한다.
 - . ← 의 경우 **현재 위치**를 의미 합니다.
 - .. ← 의 경우 **상위의 디렉토리**를 의미합니다.
 - Ex) **cd ../..** 인 경우에는 **2칸 상위**로 이동하는 것을 의미.
현재 위치가 proj/Lv01-02/TaeinPark 라고 가정
위의 명령을 입력하면 proj 로 이동 됩니다.

2. 리눅스 명령어(2)

- pwd → password 의 약자인데 사실상 현재 디렉토리의 위치를 보여줍니다.
(가장 와닿지 않는 명령어중 하나)
여기서 ~ 은 ‘/home/계정명’ 을 의미 합니다.
그래서 결국 cd ~ 을 입력하고 pwd 명령을 입력하면 ‘/home/계정명’ 이 보입니다.
*** 결국 **물결(~)**은 **사용자 계정의 홈 디렉토리**를 의미합니다. ***

```
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ pwd
/home/taein/proj/es02/Lv01-02/TaeinPark/homework/c/01
```

- sudo → 윈도우로 치면 **관리자 권한 실행**을 의미 합니다.
권한이 없다고 나오면 sudo를 입력합니다.
sudo su의 경우 관리자 권한으로 로그인을 하므로 모든 접근이 가능하지만
주의하지 않을 경우 디렉토리 설정이 꼬이면서 문제가 발생 할 수 있습니다.
- adduser → **사용자 계정 추가**
사용법) sudo adduser 계정명(test)
비밀번호 입력 후 엔터 엔터 엔터 ...
이후 su 계정명을 통해 로그인 가능
cd ~ 을 하고 pwd를 하면 위에서 설명했던
/home/ 계정명 인 /home/test 를 확인 할 수 있다.

2. 리눅스 명령어(3)

- vi → 리눅스 터미널에서 사용하는 편집기
사용법) vi 편집할 파일명
shift + l : 커서 맨 앞으로 이동하기
shift + a : 커서 맨 뒤로 이동하기
- gcc → 리눅스 전용 컴파일러 모음중 **C 컴파일러**에 해당함
사용법 1) gcc 컴파일할코드
실행시 ./a.out (여기서 a.out은 실행 파일명)
사용법 2) gcc -o 실행파일명 컴파일할코드
실행시 ./실행파일명

```
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ gcc -o a.out data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
a.out  data_type.c
```

```
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ gcc -o data_type data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
a.out  data_type  data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ rm a.out
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
data_type  data_type.c
```

- rm → 삭제 명령어
사용법 1) rm 삭제할파일명
사용법 2) rm -rf 디렉토리명
주의할 점) rm -rf /
이렇게 하면 재설치 해야함 ^^;
돌리면 1분 내로 컴퓨터가 알아서 정지 하는 것을 볼 수 있음.

3. C언어 데이터 타입

1비트 → 컴퓨터가 표현 할 수 있는 최소 단위 (0 혹은 1)

8비트 → 1 바이트

1비트가 표현 할 수 있는 숫자는 ? 0과 1로 두 가지 : 2^1

2비트가 표현 할 수 있는 숫자는 ? 00, 01, 10, 11로 네 가지 : 2^2

3비트가 표현 할 수 있는 숫자는 ? 000, 001, 010, 011, 100, 101, 110, 111로 여덟가지 : 2^3

Int - 4바이트 (32비트) : $-2^{31} \sim (2^{31} - 1)$

↳ 표현 개수가 2^{32} 에서 빼기 1이 발생하는 이유는 0을 포함하기 때문.

Short - 2바이트(16비트) : $-2^{15} \sim (2^{15} - 1)$

Char - 1바이트(8비트) : $-2^7 \sim (2^7 - 1)$

Long long int - 8바이트(64비트) : $-2^{63} \sim (2^{63} - 1)$

Float - 4바이트 :

IEEE 표준을 따르고 있으며 이 내용은 union 수업 시 고속 계산에 활용하는 케이스로 설명 예정.

Double - 8바이트

Long double - 16바이트(128비트)

4. C언어 데이터 타입 + vim 편집기 사용법(1)

```
1 #include <stdio.h>
2
3 // void main(void) <<< 굉장히 안좋은 습관(특히 리눅스)
4 // 이유는 레벨 2 리눅스 시스템 프로그래밍 할 때 설명.
5 // 프로세스(task)를 제어 할 때 return 값이 결국 프로세스가 정상 종료 했는지 비정상 종료했는지 상태값을 의미합니다.
6 // 그러나 void main(void)인 경우 이 프로세스가 정상 종료되었는지 비정상 종료 되었는지 알 수가 없습니다.
7 // 그래서 void main(void)를 권장하지 않습니다.
8
9 int main(void)
10 {
11     int num1 = 1;
12     short num2 = 2;
13     char num3 = 3;
14
15     float num4 = 2.3f;
16     double num5 = 2.8;
17
18     long long int num6 = 10000000000;
19     long double num7 = 2942723.2234;
20
21     printf("num1 = %d\n", num1);
22     printf("num2 = %d\n", num2);
23     printf("num3 = %d\n", num3);
24
25     // yy: yank -> p : paste
26     // u: 뒤로가기
27     // ctrl + r : 앞으로 가기
28     printf("num4 = %f\n", num4);
29     printf("num4 = %.2f\n", num4);
30     printf("num5 = %1f\n", num5);
31
32     return 0;
33 }
34
35 // y숫자y -> 숫자만큼 라인을 복사합니다.
36 // dd -> 삭제하기
37 // d숫자d -> 숫자만큼 라인을 삭제합니다.
38 // :$ -> 무조건 마지막 라인으로 이동합니다.
39 // :set nu -> 현재 편집기의 라인을 숫자로 보여 줍니다
40 // :set nonu -> 편집기에 보이던 숫자를 끕니다.
-- 끼워넣기 --
```

참고) 편집을 시작 할 때 'a'키를 눌러야 편집이 시작 되고, 편집을 종료하고 명령어를 할 때는 'ESC'키 입력 한다.
그리고 저장 후 종료 명령은 :wq!

4. C언어 데이터 타입 + vim 편집기 사용법(2)

```
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ vi data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ gcc data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
a.out  data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ./a.out
num1 = 1
num2 = 2
num3 = 3
num4 = 2.300000
num4 = 2.30
num5 = 2.800000
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ gcc -o a.out data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
a.out  data_type.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ./a.out
num1 = 1
num2 = 2
num3 = 3
num4 = 2.300000
num4 = 2.30
num5 = 2.800000
```


4. C언어 데이터 타입 + vim 편집기 사용법(3)

```
#include <stdio.h>

int main(void)
{
    // %d를 %lu로 바꾸라는 워닝 메시지가 발생함
    // 7개를 바꿔야 하는데 일일이 바꾸기 귀찮음 어떻게 해야 할까?
    // :%s/%d/%lu/g -> 모든 %d를 찾아서 %lu로 치환해 주세요.
    //           모든 -> /g
    //           치환 -> :%s
    //           치환 대상 -> %d
    //           치환 결과 -> %lu
    // sizeof()는 데이터 타입의 크기를 바이트 단위로 표현해 줍니다.

    printf("sizeof(char) = %lu\n", sizeof(char));
    printf("sizeof(short) = %lu\n", sizeof(short));
    printf("sizeof(int) = %lu\n", sizeof(int));
    printf("sizeof(long long int) = %lu\n", sizeof(long long int));

    printf("sizeof(float) = %lu\n", sizeof(float));
    printf("sizeof(double) = %lu\n", sizeof(double));
    printf("sizeof(long double) = %lu\n", sizeof(long double));

    return 0;
}
```

```
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ vi data_type_byte.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ gcc -o data_type_byte data_type_byte.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ls
data_type  data_type.c  data_type_byte  data_type_byte.c
taein@taein-Lenovo-ideapad-700-15ISK:~/proj/es02/Lv01-02/TaeinPark/homework/c/01$ ./data_type_byte
sizeof(char) = 1
sizeof(short) = 2
sizeof(int) = 4
sizeof(long long int) = 8
sizeof(float) = 4
sizeof(double) = 8
sizeof(long double) = 16
```