

10일차
마지막 진도

cmakelists.txt를 통해 .deb설치는 아니지만 프로젝트 관리까지는 진도를 나갔다
표준 입출력 함수에서 버퍼를 염두해야한다.

FILE * stdout / stdin	
입력	출력
write()	read()
fwrite() -> 버퍼가 꽉 차있으면 Block하는 기능이 있음	fread() -> 버퍼에 내용이 없으면 Block하는 기능이 있음 -> + 입력대기
버퍼에서의 입력	버퍼에서의 출력
printf	scanf
fprintf	fscanf
sprintf	sscanf
매크로문	
putc	getc
puts	gets
매크로문	
fputc = putchar() [stdin]	fgetc = getchar() [stdout]
fputs	fgets

****stdout 버퍼에서 중요한 것****

fflush()

버퍼에 미처리 된 것은 보냄으로써 비운다.

배포시 중요한 점

환경변수

PATH의 환경변수 확인

```
$ echo $PATH
```

환경변수 지정

```
$ set MY_VAR='test string'
```

환경변수 영구지정

```
$ export MYVAR='test string' ->$echo MYVAR 하면 뜸
```

안 쓸 때

```
$ unset MYVAR
```

환경변수 내역 확인

```
../.bashrc
```

git에 dotenv-c를 검색하면 c에서 환경변수를 라이브러리로 가져올 수 있도록 사람들이 만들어둔게 있다.

그중에 <https://github.com/lsty001/dotenv-c> 이양반 추천한다

env 가상환경 세팅된 test 파일도 있어서 보기 좋다.

실행 파일이 아닌 라이브러리(.h .o)를 만드는 용도로 cmakelists.txt를 만들었다.

git에 dotenv-c를 검색하면 c에서 환경변수를 라이브러리로 가져올 수 있도록 사람들이 만들어둔게 있다.

그중에 <https://github.com/Isty001/dotenv-c> 이양반 추천한다
env 가상환경 세팅된 test 파일도 있어서 보기 좋다.

우리 프로젝트 파일에서 git 링크를 올려서 긁어와라

```
aa@aa-virtual-machine:~/Big2025$ git clone https://github.com/Isty001/dotenv-c.git
```

긁어오면 리스트에 dotenv-c 가 생길건데 들어가봐라

```
$ cd dotenv-c/
```

여기에 ls을 쳐보면 build가 없을거니 해당 폴더를 만들어서 들어가라

```
Big2025/dotenv-c$ mkdir build
```

```
Big2025/dotenv-c$ cd build/
```

빌드 파일 안에서 cmake .. (찍어쓰기 있음 유의)

```
/Big2025/dotenv-c/build$ cmake ..
```

된거에 (빌드 뒤에 찍어쓰고 점)

```
Big2025/dotenv-c/build$ cmake --build .
```

```
Big2025/dotenv-c/build$ sudo make install
```

하면 비번 치라는데 우분투 비번

```
Consolidate compiler generated dependencies of target dotenv
```

```
[ 50%] Built target dotenv
```

```
Consolidate compiler generated dependencies of target dotenv-s
```

```
[100%] Built target dotenv-s
```

```
Install the project...
```

```
-- Install configuration: "RELEASE"
```

```
-- Installing: /usr/local/bin/libdotenv.so
```

```
-- Installing: /usr/local/include/dotenv.h
```

```
-- Installing: /usr/local/lib/libdotenv-s.a
```

```
-- Up-to-date: /usr/local/include/dotenv.h
```

라는 내용이 뜨면 잘된 것.

해석하자면 저 양반이 만든 오브젝트 파일(.o), 헤더파일(.h)이

Installing에서 Up-to-date로 넘어갔다는거다.

