



# fd

## File Descriptor

프로세스가 파일에 접근할 때 사용되는 인덱스

### c함수 파일 읽기 read()

- 형태 : `ssize_t read(int fd, void *buf, size_t nbytes)`
- 인수 : `int fd`            파일 디스크립터(**fd**)  
          `void * buf`        파일을 읽어 들일 버퍼  
          `size_t nbytes`   버퍼의 크기
- 반환 : 정상적으로 실행되었다면 읽어들이는 바이트 수

### fd 값의 종류

- 0 == 표준입력(키보드)
- 1 == 표준 출력(모니터)
- 2 == 표준 에러(모니터)

#### ▼ 그 외 함수

##### 문자열 비교 strcmp

- 형태 : `int strcmp(const char* str1, const char* str2)`
- 인자 : `str1` 비교할 문자열 1  
          `str2` 비교할 문자열 2
- 반환 : `str1 < str2` 인 경우, 음수 반환  
          `str1 > str2` 인 경우, 양수 반환  
          `str1 == str2` 인 경우, 0 반환

##### 형변환 함수 atoi

- 형태 : `int atoi(const char* cStr)`

- 문자열을 정수 타입으로 변환해서 반환

풀이

putty를 이용해 pwnable.kr에 접속

```
pwnable.kr - PuTTY
Using username "fd".
fd@pwnable.kr's password:
pwnable.kr
- Site admin : daehee87@gatech.edu
- IRC : irc.netgarage.org:6667 / #pwnable.kr
- Simply type "irssi" command to join IRC now
- files under /tmp can be erased anytime. make your directory under /tmp
- to use peda, issue `source /usr/share/peda/peda.py` in gdb terminal
You have mail.
Last login: Fri Aug 13 08:48:41 2021 from 5.29.59.201
fd@pwnable:~$
```

`ls -l` 를 이용해 파일들을 살펴보자

```
pwnable.kr - PuTTY
Using username "fd".
fd@pwnable.kr's password:
[ASSEMBLY]
- Site admin : daehee87@gatech.edu
- IRC : irc.netgarage.org:6667 / #pwnable.kr
- Simply type "irssi" command to join IRC now
- files under /tmp can be erased anytime. make your directory under /tmp
- to use peda, issue `source /usr/share/peda/peda.py` in gdb terminal
You have mail.
Last login: Fri Aug 13 08:48:41 2021 from 5.29.59.201
fd@pwnable:~$ ls -l
total 16
-r-sr-x--- 1 fd_pwn fd 7322 Jun 11 2014 fd
-rw-r--r-- 1 root root 418 Jun 11 2014 fd.c
-r--r----- 1 fd_pwn root 50 Jun 11 2014 flag
```

우리는 지금 fd로 로그인했기 때문에 fd파일에 대해서 읽고 쓰는것이 가능하고,  
fd.c파일에서는 user와 group이 root이기 때문에 우리는 other로 분류되서 읽는 것 밖에 허  
용이 안된다.

cat fd.c 로 fd.c파일을 살펴보자

```
pwnable.kr - PuTTY
fd@pwnable:~$ cat fd.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char buf[32];
int main(int argc, char* argv[], char* envp[]){
    if(argc<2){
        printf("pass argv[1] a number\n");
        return 0;
    }
    int fd = atoi( argv[1] ) - 0x1234;
    int len = 0;
    len = read(fd, buf, 32);
    if(!strcmp("LETMEWIN\n", buf)){
        printf("good job :)\n");
        system("/bin/cat flag");
        exit(0);
    }
    printf("learn about Linux file IO\n");
    return 0;
}
```

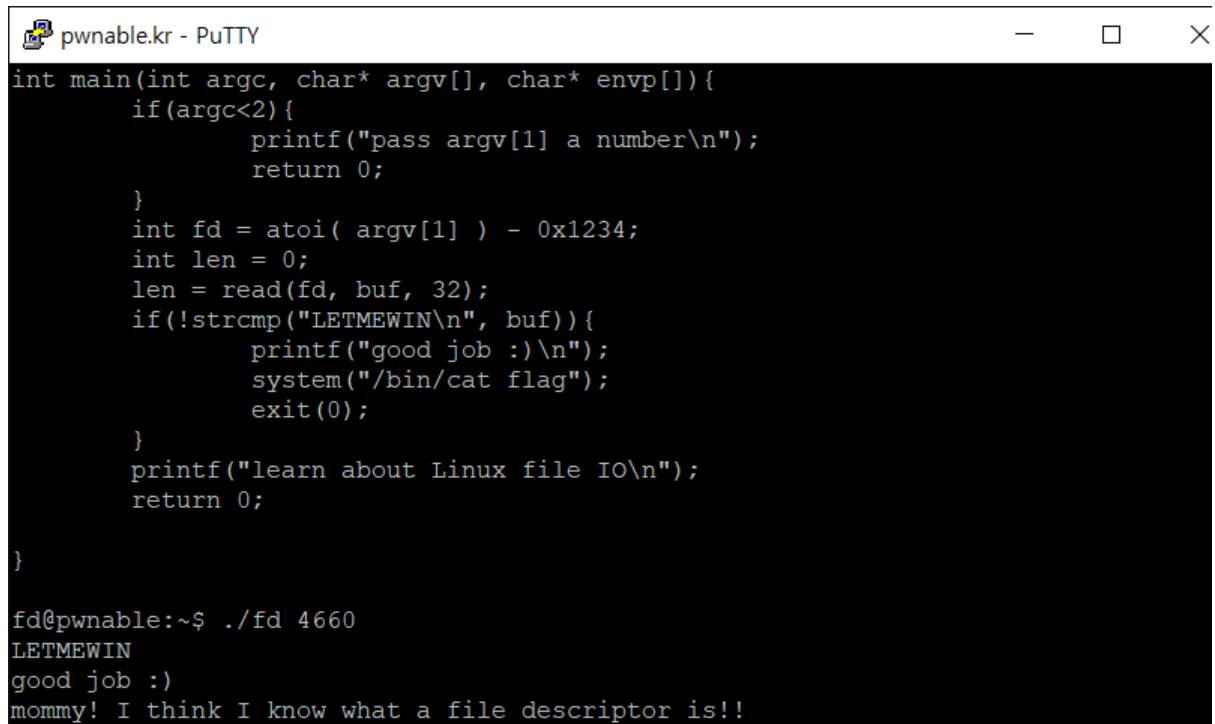
## ▼ main의 인자

int argc : 메인함수에 전달되는 정보의 갯수

char\* argv[ ] : 메인함수에 전달되는 실직적인 정보로, 문자열의 배열을 의미

main함수의 if문을 보니 인자를 하나이상 넣으라는 것을 의미한다

위에서 read()함수를 보면 알듯 fd의 값을 0으로 줘서 buf의 값을 LETMEWIN\n으로 넣어 주면 답은 풀린다. fd 값을 0으로 만드려면 main함수에 0x1234의 값(10진수로 4660)을 넘겨주면 된다. 따라서 `./fd 4660` 을 입력하고 LETMEWIN을 입력해주면 flag를 알수있다.



```
int main(int argc, char* argv[], char* envp[]){
    if(argc<2){
        printf("pass argv[1] a number\n");
        return 0;
    }
    int fd = atoi( argv[1] ) - 0x1234;
    int len = 0;
    len = read(fd, buf, 32);
    if(!strcmp("LETMEWIN\n", buf)){
        printf("good job :)\n");
        system("/bin/cat flag");
        exit(0);
    }
    printf("learn about Linux file IO\n");
    return 0;
}

fd@pwnable:~$ ./fd 4660
LETMEWIN
good job :)
mommy! I think I know what a file descriptor is!!
```