Standard I/O Library

System Programming

실습

gcc 사용하기

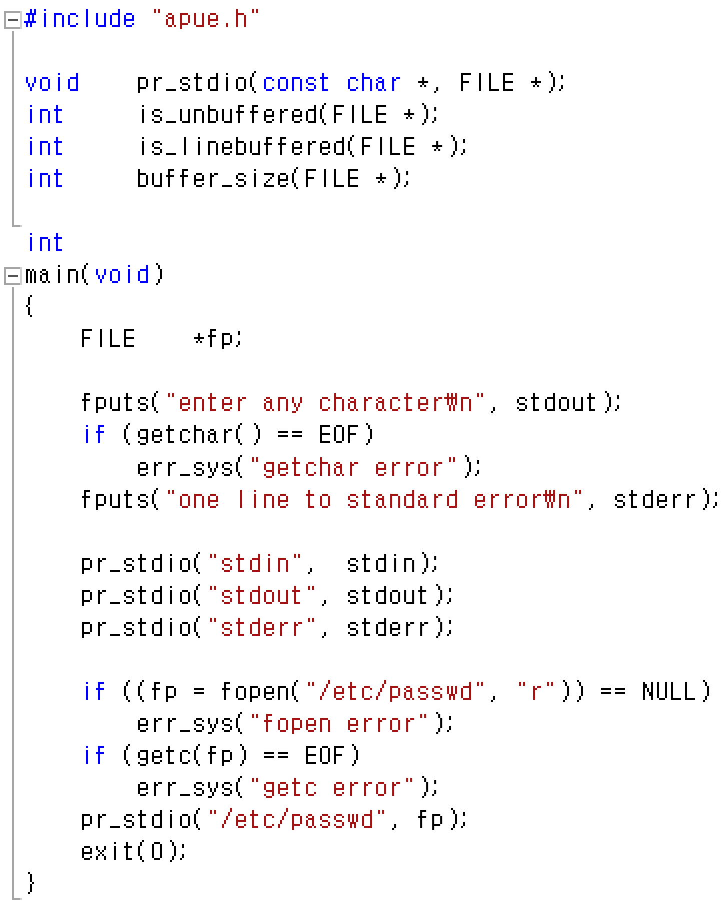
❑ 책자(이론) 예제 코드 컴파일 및 실행 해보기

- Prog. 5.11
  - 기본 예제 코드를 수행 후, /etc/passwd파일에 대해서 buffering방식을 바꿔보기
  - setvbuf함수를 이용해서 줄단위 버퍼링과, 버퍼링 없는 방식으로 동작하도록 수정해보기
- Prog. 5.12
- Prog. 5.13

실습

StdIO system call

❑ Prog. 5. 11 코드

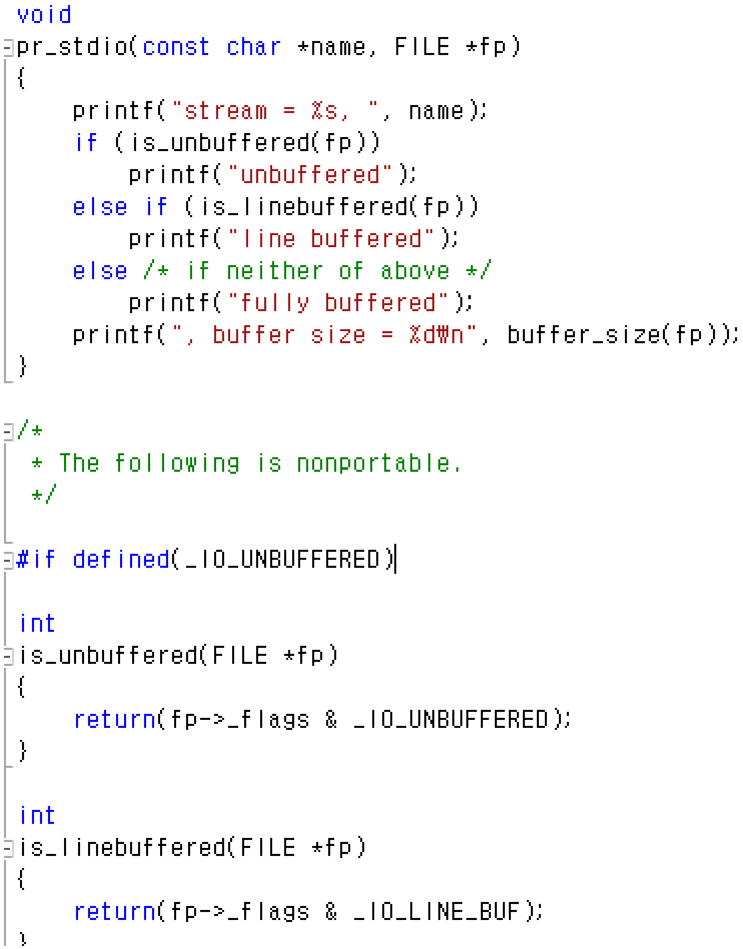


```
#include "apue.h"

void    pr_stdio(const char *, FILE *);
int     is_unbuffered(FILE *);
int     is_linebuffered(FILE *);
int     buffer_size(FILE *);

int
main(void)
{
    FILE    *fp;

    fputs("enter any character₩n", stdout);
    if (getchar() == EOF)
        err_sys("getchar error");
    fputs("one line to standard error₩n", stderr);

    pr_stdio("stdin",  stdin);
    pr_stdio("stdout", stdout);
    pr_stdio("stderr", stderr);

    if ((fp = fopen("/etc/passwd", "r")) == NULL)
        err_sys("fopen error");
    if (getc(fp) == EOF)
        err_sys("getc error");
    pr_stdio("/etc/passwd", fp);
    exit(0);
}
```

실습

StdIO system call

❑ Prog. 5. 11 코드

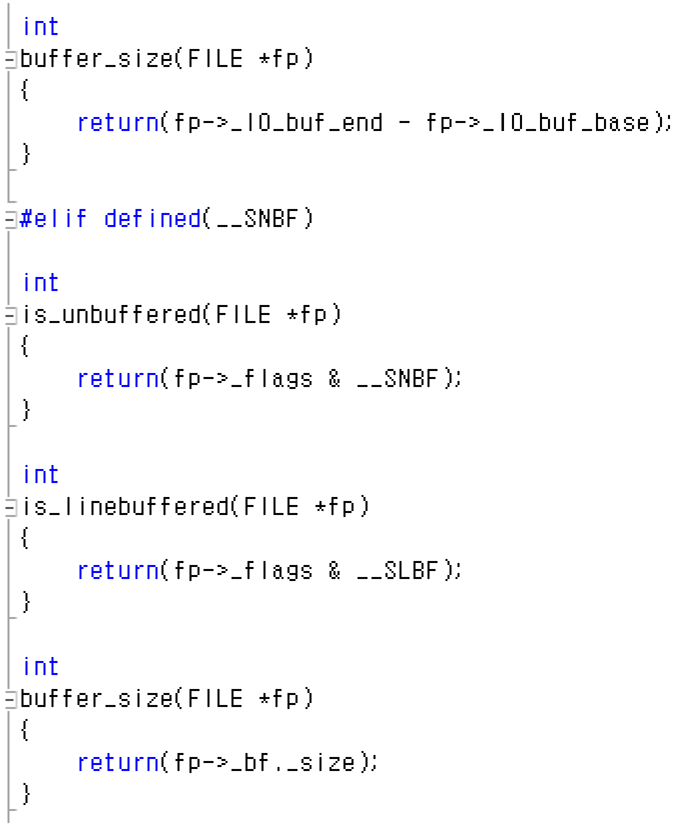


```
void
pr_stdio(const char *name, FILE *fp)
{
    printf("stream = %s, ", name);
    if (is_unbuffered(fp))
        printf("unbuffered");
    else if (is_linebuffered(fp))
        printf("line buffered");
    else /* if neither of above */
        printf("fully buffered");
    printf(", buffer size = %d₩n", buffer_size(fp));
}

/*
 * The following is nonportable.
 */

#if defined(_IO_UNBUFFERED)

int
is_unbuffered(FILE *fp)
{
    return(fp->_flags & _IO_UNBUFFERED);
}

int
is_linebuffered(FILE *fp)
{
    return(fp->_flags & _IO_LINE_BUF);
}
```

실습

StdIO system call

❑ Prog. 5. 11 코드

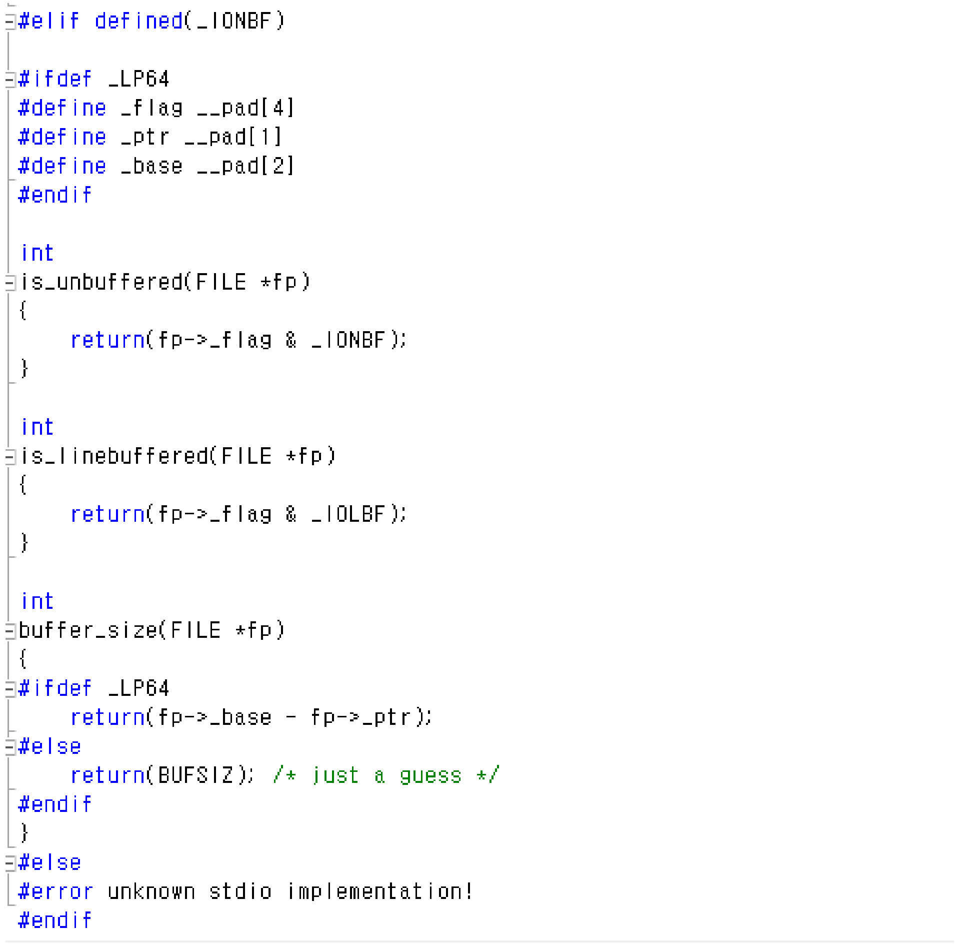


```
int
buffer_size(FILE *fp)
{
	return(fp->_IO_buf_end - fp->_IO_buf_base);
}

#elif defined(__SNBF)

int
is_unbuffered(FILE *fp)
{
	return(fp->_flags & __SNBF);
}

int
is_linebuffered(FILE *fp)
{
	return(fp->_flags & __SLBF);
}

int
buffer_size(FILE *fp)
{
	return(fp->_bf._size);
}
```

실습

StdIO system call

❑ Prog. 5. 11 코드

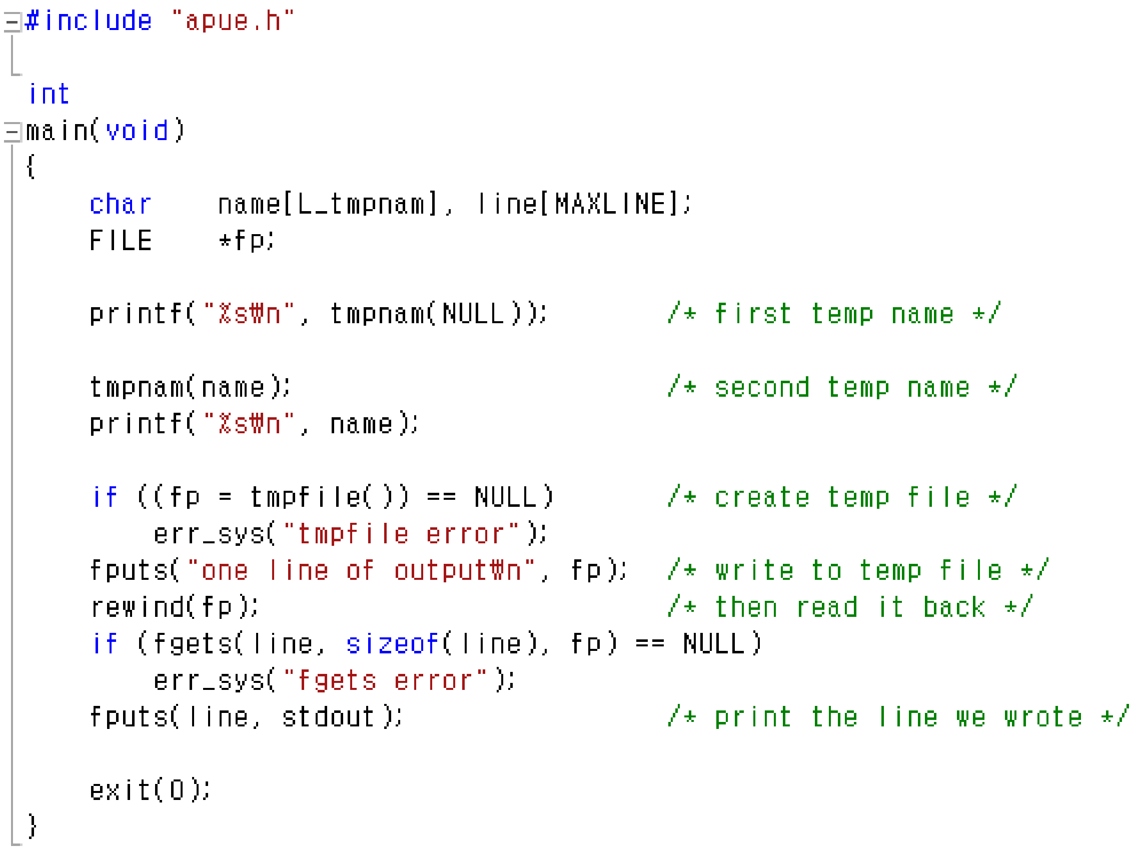


```
#elif defined(_IONBF)

#ifdef _LP64
#define _flag __pad[4]
#define _ptr __pad[1]
#define _base __pad[2]
#endif

int
is_unbuffered(FILE *fp)
{
    return(fp->_flag & _IONBF);
}

int
is_linebuffered(FILE *fp)
{
    return(fp->_flag & _IOLBF);
}

int
buffer_size(FILE *fp)
{
#ifdef _LP64
    return(fp->_base - fp->_ptr);
#else
    return(BUFSIZ); /* just a guess */
#endif
}
#else
#error unknown stdio implementation!
#endif
```

실습

StdIO system call

❑ Prog. 5. 12 코드

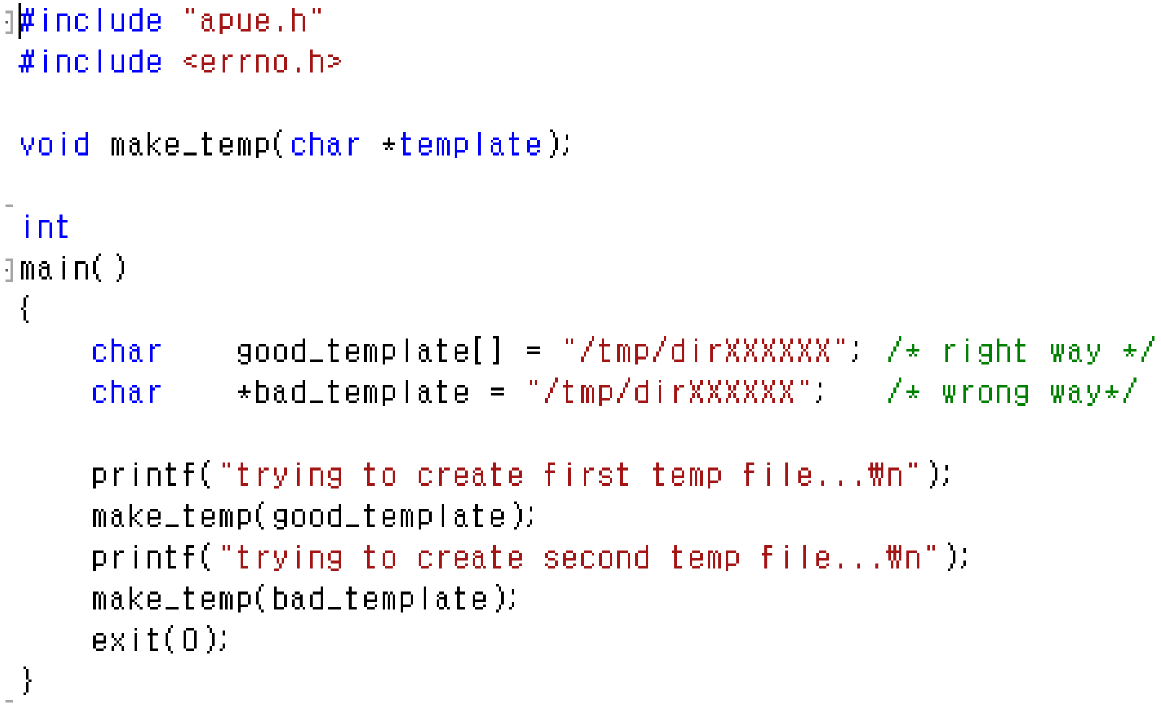


```
#include "apue.h"

int
main(void)
{
    char    name[L_tmpnam], line[MAXLINE];
    FILE    *fp;

    printf("%s₩n", tmpnam(NULL));      /* first temp name */

    tmpnam(name);                      /* second temp name */
    printf("%s₩n", name);

    if ((fp = tmpfile()) == NULL)      /* create temp file */
        err_sys("tmpfile error");
    fputs("one line of output₩n", fp);  /* write to temp file */
    rewind(fp);                        /* then read it back */
    if (fgets(line, sizeof(line), fp) == NULL)
        err_sys("fgets error");
    fputs(line, stdout);               /* print the line we wrote */

    exit(0);
}
```

실습

StdIO system call

❑ Prog. 5. 13 코드

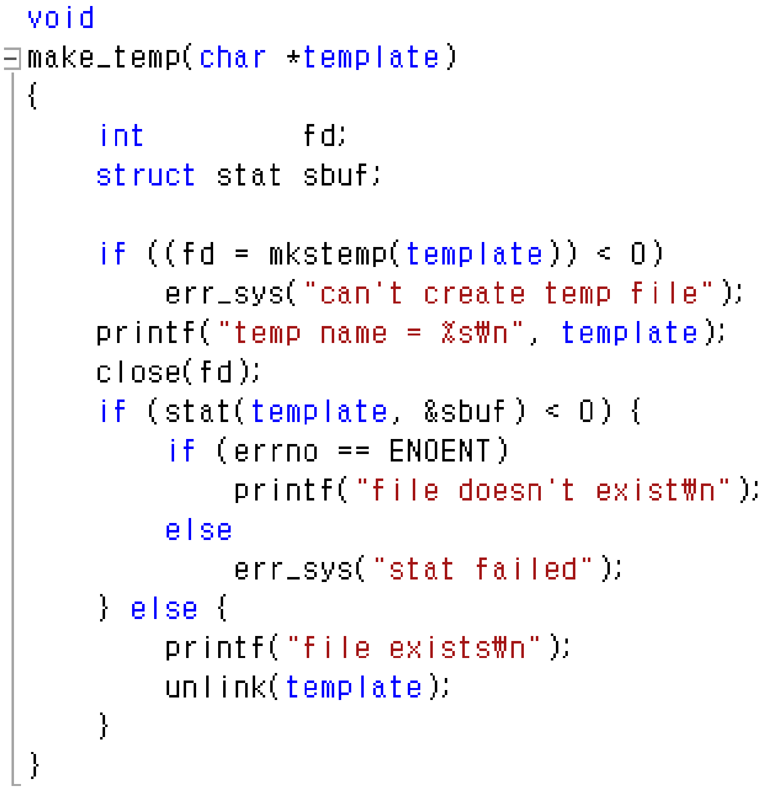


```
#include "apue.h"
#include <errno.h>

void make_temp(char *template);

int
main()
{
        char    good_template[] = "/tmp/dirXXXXXX"; /* right way */
        char    *bad_template = "/tmp/dirXXXXXX";   /* wrong way*/

        printf("trying to create first temp file...₩n");
        make_temp(good_template);
        printf("trying to create second temp file...₩n");
        make_temp(bad_template);
        exit(0);
}
```

실습

StdIO system call

❑ Prog. 5. 13 코드

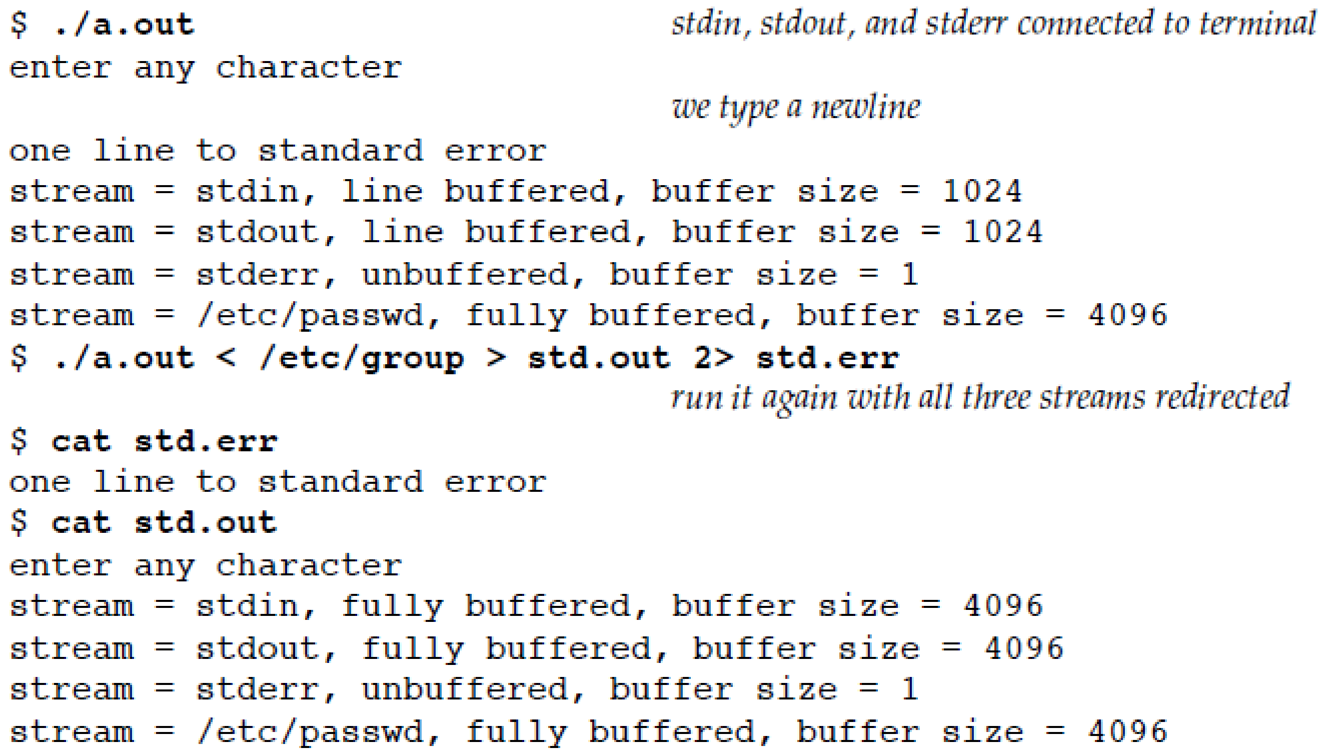


```
void
make_temp(char *template)
{
    int         fd;
    struct stat sbuf;

    if ((fd = mkstemp(template)) < 0)
        err_sys("can't create temp file");
    printf("temp name = %s₩n", template);
    close(fd);
    if (stat(template, &sbuf) < 0) {
        if (errno == ENOENT)
            printf("file doesn't exist₩n");
        else
            err_sys("stat failed");
    } else {
        printf("file exists₩n");
        unlink(template);
    }
}
```

실습

StdIO system call

❑ Prog. 5. 11 실행

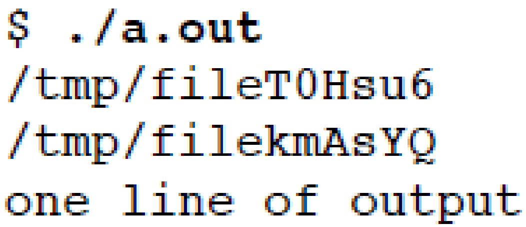


```
$ ./a.out                              stdin, stdout, and stderr connected to terminal
enter any character
                                       we type a newline
one line to standard error
stream = stdin, line buffered, buffer size = 1024
stream = stdout, line buffered, buffer size = 1024
stream = stderr, unbuffered, buffer size = 1
stream = /etc/passwd, fully buffered, buffer size = 4096
$ ./a.out < /etc/group > std.out 2> std.err
                                       run it again with all three streams redirected
$ cat std.err
one line to standard error
$ cat std.out
enter any character
stream = stdin, fully buffered, buffer size = 4096
stream = stdout, fully buffered, buffer size = 4096
stream = stderr, unbuffered, buffer size = 1
stream = /etc/passwd, fully buffered, buffer size = 4096
```

실습

StdIO system call

❑ **Prog. 5. 12 실행**

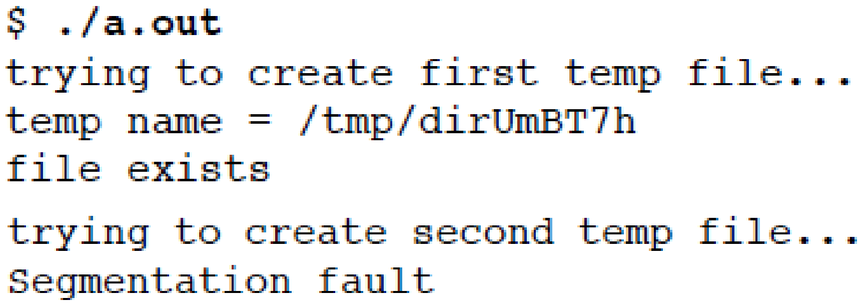


```
$ ./a.out
/tmp/fileT0Hsu6
/tmp/filekmAsYQ
one line of output
```

실습

StdIO system call

- ❑ **Prog. 5. 13 실행**



```
$ ./a.out
trying to create first temp file...
temp name = /tmp/dirUmBT7h
file exists
trying to create second temp file...
Segmentation fault
```

Thank you for your attention !!

Q and A