O_APPEND

- open이 성공하면 파일의 마지막 바이트 바로 뒤에 위치
- 그 이후의 write는 전부 파일의 끝에 자료를 추가하게 됨
- 파일의 끝에 자료를 추가하는 방법
  - lseek 사용

```
lseek(fd, 0, SEEK_END)
write(fd, buf, BUFSIZE)
```

  - O_APPEND

```
open( “filename” , O_WRONLY|O_APPEND)
write(fd, buf, BUFSIZE)
```

표준 입력, 표준 출력 및 표준 오류

표준 입력(0), 표준 출력(1), 표준 오류(2)

- redirection < >
  - prog_name < infile
    - 파일기술자 0로부터 읽어 들일 때, infile로 부터 자료를 읽음
  - prog_name > outputfile
    - 출력을 outputfile로 변경
  - prog_name < infile > outputfile
- pipe
  - prog1 | prog2

test3.c

예제

```
01  #include <fcntl.h>
02  #include <unistd.h>
03  #include <stdlib.h>
04  #include <stdio.h>
05  #define SIZE 512

06  int main(void) {
07      ssize_t nread;
08      char buf[SIZE]
09
10      while ((nread = read(0, buf, SIZE)) > 0
11              write(1, buf, nread);
12
13      return 0;
14  }
```

실습

- [문제1]위 프로그램을 수정하여 명령줄 인수가 있는지 조사하고, 만일 인수가 존재하면, 각 인수를 하나의 파일 이름으로 취급하고 각 파일의 내용을 표준 출력으로 복사하는 프로그램을 작성하시오. 만일 명령 줄 인수가 존재하지 않으면, 입력을 표준 입력으로부터 받아야 한다.

- [문제2][문제1]의 표준출력부분을 outfile에 저장되도록 실행명령을 수정하시오.

파일 기술자 제어

파일 기술자 제어 : fcntl(2)

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
int fcntl(int fildes, int cmd, /* arg */ ...);
```

- 파일 기술자가 가리키는 파일에 cmd로 지정한 명령을 수행
- cmd의 종류에 따라 인자(arg)를 지정할 수 있음
- 자주 사용하는 cmd

F_GETFL	상태 플래그 정보를 읽어온다.
F_SETFL	상태 플래그 정보를 설정한다. 설정할 수 있는 플래그는 대부분 open 함수에서 지정하는 플래그다.

[예제 2-9] fcntl 함수로 파일 기술자 제어하기

```
07 int main(void) {
08      int fd, flags;
09
10      fd = open("unix.txt", O_RDWR);
11      if (fd == -1) {
12          perror("open");
13          exit(1);
14      }
15
16      if ((flags = fcntl(fd, F_GETFL)) == -1) {
17          perror("fcntl");
18          exit(1);
19      }
20
21      flags |= O_APPEND;
22
23      if (fcntl(fd, F_SETFL, flags) == -1) {
24          perror("fcntl");
25          exit(1);
26      }
27
28      if (write(fd, "Hanbit Media", 12) != 12) perror("write");
29      close(fd);
30
31      return 0;
32  }
```

파일을 추가 모드로 수정

파일에 내용 추가

```
# cat unix.txt
Unix System Programming
# ex2_9.out
# cat unix.txt
Unix System Programming
Hanbit Media
```

실습

- 위 프로그램을 수정하여 현재의 파일 플래그를 테스트하여 출력하고, O_APPEND를 추가하는 프로그램을 작성하라

```
switch (현재 파일플래그 & O_ACCMODE) {
    case O_RDWR :
    ...
}
```

파일 삭제

unlink(2)

```
#include <unistd.h>
int unlink(const char *path);
```

- path에 지정한 파일의 inode에서 링크 수를 감소시킨다.
- 링크 수가 0이 되면 path에 지정한 파일이 삭제된다.
- 파일 뿐만 아니라 디렉토리(빈 디렉토리 아니어도 됨)도 삭제된다.

remove(3)

```
#include <stdio.h>
int remove(const char *path);
```

- path에 지정한 파일이나 디렉토리를 삭제한다.
- 디렉토리인 경우 빈 디렉토리만 삭제한다.

fsync – 메모리에 위치하고 있는 파일의 내용을 디스크로 보내 메모리와 디스크의 내용을 동기화한다. 메모리의 내용이 디스크로 모두 기록되기 전에는 리턴하지 않는다

int fsync(int filedes);