

파일 오프셋 지정

□ 파일 오프셋 위치 지정 : lseek(2) 함수 다 제시해줌.

```
#include <sys/types.h>
#include <unistd.h>
off_t lseek(int fildes, off_t offset, int whence);
```

- offset으로 지정한 크기만큼 오프셋을 이동시킨다.
- offset의 값은 whence값을 기준으로 해석한다.

```
lseek(fd, 5, SEEK_SET);
lseek(fd, 0, SEEK_END);
```

파일의 시작에서
5번째 위치로 이동

파일의 끝에서
0번째, 즉 끝으로 이동

★ 플래그는 외워야 함!!!

값	설명
SEEK_SET	<u>파일의 시작 기준</u>
SEEK_CUR	<u>현재 위치 기준</u>
SEEK_END	<u>파일의 끝 기준</u>

파일의 사이즈를 알 수 있음.

- 파일 오프셋의 현재 위치를 알려면?

```
cur_offset = lseek(fd, 0, SEEK_CUR);
```



```
...
07  int main(void) {
08      int fd, n;
09      off_t start, cur;
10      char buf[256];
11
12      fd = open("unix.txt", O_RDONLY);
13      if (fd == -1) {
14          perror("Open unix.txt");
15          exit(1);
16      }
17
18      start = lseek(fd, 0, SEEK_CUR);
19      n = read(fd, buf, 255);
20      buf[n] = '\0';
21      printf("Offset start=%d, Read Str=%s, n=%d\n", (int)start, buf, n);
22      • cur = lseek(fd, 0, SEEK_CUR);
23      printf("Offset cur=%d\n", (int)cur);
24
```



[예제 2-6] 파일 오프셋 사용하기

```
25     start = lseek(fd, 5, SEEK_SET);
26     n = read(fd, buf, 255);
27     buf[n] = '\0';
28     printf("Offset start=", Read Str=%s", (int)start, buf);
29
30     close(fd);
31
32     return 0;
33 }
```

```
# ex2_6.out
Offset start=0, Read Str=Unix System Programming
, n=24
Offset cur=24
Offset start=5, Read Str=System Programming
```



- createdata.c 실행 (residents 파일 생성)
- 실습 4문제는 수업시간에 upload
- [문제 4]는 이번 주 일요일(9/12)까지 제출할 것



파일 기술자 복사

□ 파일 기술자 복사 : dup(2)

```
#include <unistd.h>
int dup(int fildes);
```

- 기존 파일 기술자를 인자로 받아 새로운 파일 기술자를 리턴
- 새로운 파일 기술자는 현재 할당할 수 있는 파일 기술자 중 가장 작은 값으로 자동 할당

□ 파일 기술자 복사 : dup2(3)

```
#include <unistd.h>
int dup2(int fildes, int fildes2);
```

- 새로운 파일 기술자를 지정할 수 있다.



```
01 #include <fcntl.h>
02 #include <unistd.h>
03 #include <stdlib.h>
04 #include <stdio.h>
05
06 int main(void) {
07     int fd, fd1;
08
09     fd = open("tmp.aaa", O_CREAT | O_WRONLY | O_TRUNC, 0644);
10     if (fd == -1) {
11         perror("Create tmp.aaa");
12         exit(1);
13     }
14
15     close(1);
16
17     fd1 = dup(fd);
18
19     printf("DUP FD=%d\n", fd1);
20     printf("Standard Output Redirection\n");
21     close(fd);
22
23     return 0;
24 }
```

표준출력(1)을 닫았다

fd를 복사하면 가장 작은
값인 1로 복사

표준출력을 출력한 내용이 파일로 저
장

ex2_7.out
cat tmp.aaa
DUP FD=1
Standard Output Redirection

```
01  #include <fcntl.h>
02  #include <unistd.h>
03  #include <stdlib.h>
04  #include <stdio.h>
05
06  int main(void) {
07      int fd;
08
09      fd = open("tmp.bbb", O_CREAT | O_WRONLY | O_TRUNC, 0644);
10      if (fd == -1) {
11          perror("Create tmp.bbb");
12          exit(1);
13      }
14
15      dup2(fd, 1);
16
17      printf("DUP2 : Standard Output Redirection\n");
18      close(fd);
19
20      return 0;
21  }
```

표준출력(1)로
지정하여 복사

표준출력을 출력한
내용이 파일로 저장된다.

```
# ex2_8.out
# cat tmp.bbb
DUP2 : Standard Output Redirection
```

