System Programming & OS 실습

6. File I/O

정지헌, 안석현, 김선재
Dankook University
{wlgjsjames7224, seokhyun, rlatjswo0824}@dankook.ac.kr

Index

- open(), Read()
- write()
- mycat
- create new file
- lseek

System call

❖ Allowing a process to request a kernel service.

❖ The primary interface between processes and the operating system, providing a means to invoke services made available by the operating system [Operating System Concepts 10th]

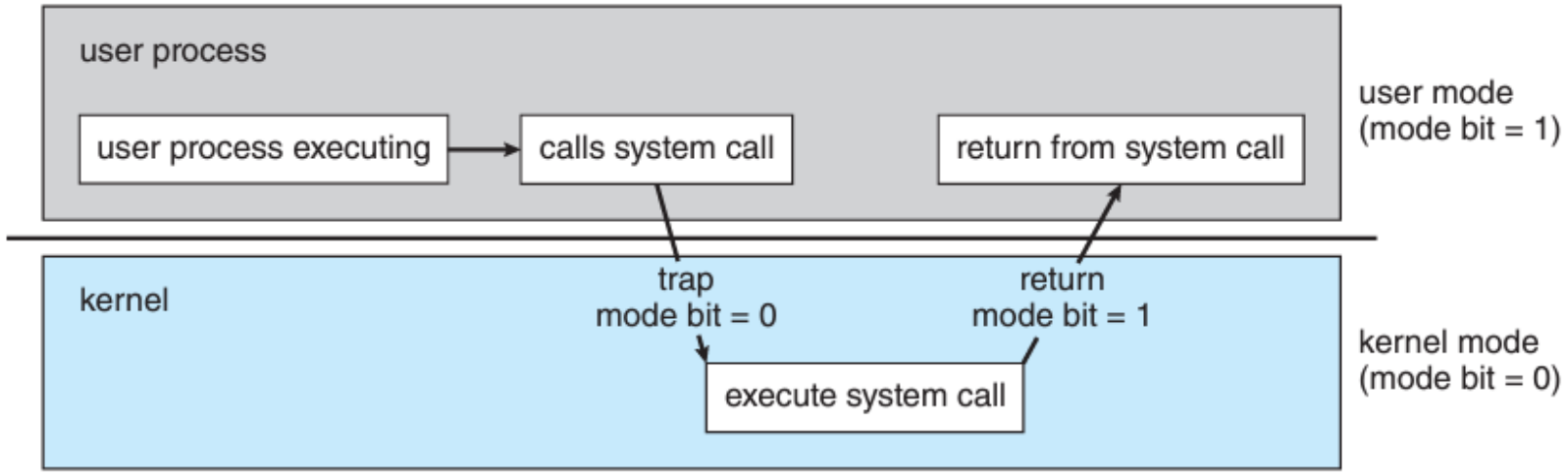

Figure 1.13 Transition from user to kernel mode.

System call

❖ Work

- File I/O, Process management, network, memory…

[Operating System Concepts 10th]

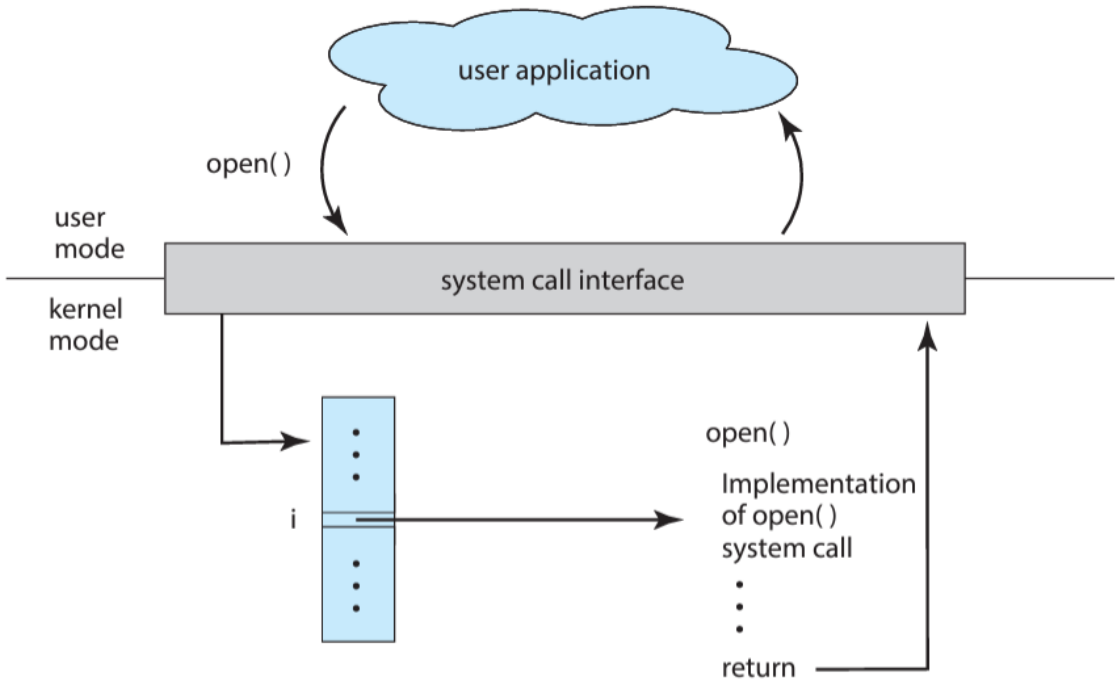

Figure 2.6 The handling of a user application invoking the `open()` system call.

open(), read()

❖ open()

- parameters(const char *pathname, int flags, mode_t mode)
  - *const char pathname: The path of the file to be opened
  - Int flags: specifies the access mode of the file

    (e.g., O_RDONLY, O_WRONLY, O_RDWR, O_CREAT: 파일 생성, O_EXCL: 파일이 존재할 시 -1 반환)
  - O_CREAT 일 시, 파라미터 mode_t mode 호출
- return value
  - On a successful read, the number of non-negative integer.
  - If an error occurs, read() return -1.

https://man7.org/linux/man-pages/index.html

open(), read()

❖ read()

- parameters(int fd, void *buf, size_t count)
  - int fd: the file descriptor to be read
  - void *buf: a buffer into which the data will be read
  - size_t count: the maximum number of bytes to be read into the buffer
- return value
  - On a successful read, the number of bytes read is returned
  - A return value of 0 indicates end of file
  - If an error occurs, read() return -1.

https://man7.org/linux/man-pages/index.html

open(), read()


```
[ec2-user@ip-172-31-15-105 ~]$ vi open.c
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <fcntl.h>

#define MAX_BUF 5
char fname[] = "alphabet.txt";

int main() {
    int fd, size;
    char buf[MAX_BUF];

    fd = open([redacted]);
    if (fd < 0) {
        printf("Can't open %s file with errno %d\n", fname, errno);
        exit(-1);
    }

    size = read([redacted]);
    if (size < 0) {
        printf("Can't read from file %s, size = %d\n", fname, size);
    } else {
        printf("size of read data is %d\n", size);
    }

    close(fd);
    return 0;
}

```

open(), read()


```
[ec2-user@ip-172-31-15-105 taba7]$ gcc -o open open.c
[ec2-user@ip-172-31-15-105 taba7]$ ls
open  open.c
[ec2-user@ip-172-31-15-105 taba7]$ ./open
Can't open alphabet.txt file with errno 2
```

파일 및 디렉토리 x

Linux Error Codes

Number	Error Code	Description
1	EPERM	Operation not permitted
2	ENOENT	No such file or directory
3	ESRCH	No such process
4	EINTR	Interrupted system call

open(), read()


```
[ec2-user@ip-172-31-15-105 taba7]$ vi alphabet.txt
  1 abcdefg
~
~
~
~
~
~
~
~
~
```

```
[ec2-user@ip-172-31-15-105 taba7]$ ./open
size of read data is 5
```

Write()

❖ write()

- parameters(int fd, const void *buf, size_t count)
  - int fd: the file descriptor to be write
  - void *buf: a buffer into which the data will be write
  - size_t count: the maximum number of bytes to be write into the buffer
- return value
  - On a successful read, the number of bytes write is returned
  - If an error occurs, read() return -1.

https://man7.org/linux/man-pages/index.html

Write()

write.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>

#define MAX_BUF 5
char fname[]="alphabet.txt";

int main(){
    int fd,read_size,write_size;
    char buf[MAX_BUF];

    fd = open(                    );
    if(fd<0){
        printf("Can't open %s file with errno %d\n",fname,errno);
        exit(-1);
    }
    read_size = read(                    );
    if(read_size < 0){
        printf("Can't read from file %s, size= %d\n",fname,write_size);
    }
    write_size = write(                                  );
    close(fd);
}

```

STDIN_FILENO:표준 입력
STOUT_FILENO: 표준 출력

```
[ec2-user@ip-172-31-15-105 taba7]$ gcc -o write write.c
[ec2-user@ip-172-31-15-105 taba7]$ ./write
abcde[ec2-user@ip-172-31-15-105 taba7]$
```

5개만 읽고 출력
Why?
두 가지 방법

Write_1.c
Write_2.c

main()

❖ main(int argc, char* argv)

- int argc: main함수에 전달되는 인자의 개수 + 1
- char* argv[0]: 실행된 프로그램의 경로와 프로그램 이름
- char* argv[1]: 첫번째 인자
- . 두번째 인자
- . 세번째 인자

main.c

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char* argv[]) {
    for(int i=0; i<argc; i++)
        printf("argv[%d]은  %s입 니 다 .\n", i, argv[i]);
    printf("argc는  %d개  입 니 다 .\n", argc);

    return 0;
}
```

❖ 실습) main 프로그램 실행

⇒ 인자 5개 전달 후 결과 확인

⇒ Input: *./main 1 2 3 4 5* or *[main path]/main 1 2 3 4 5*

mycat

mycat.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>

#define MAX_BUF 64

int main(int argc, char *argv[]){
    int fd,read_size,write_size;
    char buf[MAX_BUF];

    if([redacted]){
        printf("USAGE: %s file_name\n",argv[0]);
        exit(-1);
    }
    fd = open(argv[1], O_RDONLY);
    if([redacted]
        rintf("Can't open %s file with errno %d\n", argv[1], errno);
        exit(-1);
        //open error handling
    }
    while[redacted]
        read_size=read(fd,buf,MAX_BUF);
        if(read_size == 0)
            break;
        write_size=write(STDOUT_FILENO,buf,read_size);
    }
    close(fd);
}
```

Hint: Output

```
[ec2-user@ip-172-31-15-105 day6]$ ./mycat
USAGE: ./mycat file_name
[ec2-user@ip-172-31-15-105 day6]$ ./mycat alphabat.txt
Can't open alphabat.txt file with errno 2
[ec2-user@ip-172-31-15-105 day6]$ ./mycat alphabet.txt
abcdefg
```

1 2 3

Number	Error Code	Description
1	EPERM	Operation not permitted
2	ENOENT	No such file or directory
3	ESRCH	No such process
4	EINTR	Interrupted system call

Create new file

mycreat.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#define MAX_BUF 64
char fname[]="newfile.txt";
char dummy_data[]="abcdefg\n";

int main(){
    int fd,read_size,write_size;
    char buf[MAX_BUF];

    fd = open
    if(fd<0){
        printf("Can't create %s file with errno %d\n",fname,errno);
        exit(1);
    }
    write_size=write(fd,dummy_data,sizeof(dummy_data));
    printf("write_size = %d\n",write_size);
    close(fd);

    fd=open(fname,O_RDONLY);
    read_size = read(fd,buf,MAX_BUF);
    printf("read_size = %d\n",read_size);
    write_size = write(STDOUT_FILENO,buf,read_size);

    close(fd);
}
```

- 파일은 다른 사람이 수정을 못한다.

Hint 1: parameter

❖ open()

- parameters(const char *pathname, int flags, mode_t mode)
  - *const char pathname: The path of the file to be opened
  - Int flags: specifies the access mode of the file

    (e.g., O_RDONLY, O_WRONLY, O_RDWR, O_CREAT: 파일 생성, O_EXCL: 파일이 존재할 시 -1 반환)
  - O_CREAT 일 시, 파라미터 mode_t mode 호출

Hint 2:
mode_t mode 0664

Hint 3: Output

```
[ec2-user@ip-172-31-15-105 day6]$ ./mycreat
write_size = 9
read_size = 9
abcdefg
```

Create new file

mycreat_1.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#define MAX_BUF 64
char fname[]="newfile.txt";
char dummy_data[]="abcdefg\n";

int main(){
    int fd,read_size,write_size;
    char buf[MAX_BUF];

    fd = open(fname,O_RDWR | O_CREAT | O_EXCL, 0664);
    if(fd<0){
        printf("Can't create %s file with errno %d\n",fname,errno);
        exit(1);
    }
    write_size=write(fd,dummy_data,sizeof(dummy_data));
    printf("write_size = %d\n",write_size);
    close(fd);

    fd=open(fname,O_RDONLY);
    read_size = read(fd,buf,MAX_BUF);
    printf("read_size = %d\n",read_size);
    write_size = write(STDOUT_FILENO,buf,read_size);

    close(fd);
}
```

Hint: Output

```
I'm[ec2-user@ip-172-31-15-105 day6]$ ./mycreat_1
File name: Newfile.txt
Enter the data: Newfile
Newfile[ec2-user@ip-172-31-15-105 day6]$ cat Newfile.txt
Newfile[ec2-user@ip-172-31-15-105 day6]$
```

lseek()


```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#define MAX_BUF 64
char fname[]="newfile.txt";
char dummy_data[]="abcdefg\n";

int main(){
    int fd,read_size,write_size,new_offset;
    char buf[MAX_BUF];

    fd = open(fname,O_RDWR | O_CREAT | O_EXCL, 0664);
    if(fd<0){
        printf("Can't create %s file with errno %d\n",fname,errno);
        exit(1);
    }
    write_size=write(fd,dummy_data,sizeof(dummy_data));
    close(fd);

    fd=open(fname,O_RDONLY);
    new_offset = lseek([redacted]);
    read_size = read(fd,buf,MAX_BUF);
    printf("read_size = %d\n",read_size);
    write_size = write(STDOUT_FILENO,buf,read_size);

    close(fd);
}
```

Hint: Parameter

https://man7.org/linux/man-pages/man2/lseek.2.html

Hint: Output

```
[ec2-user@ip-172-31-15-105 day6]$ ./lseek
read_size = 6
defg
```

그 이외 시스템 콜

- create()
- mkdir(), readdir(), rmdir()
- pipe()
- mknod()
- link(),unlink()
- dup(),dup2()
- stat(),fstat()
- chmod(), fchmod()
- Ioctl(), fcntl()
- Sync(), fsync()