File & Directories

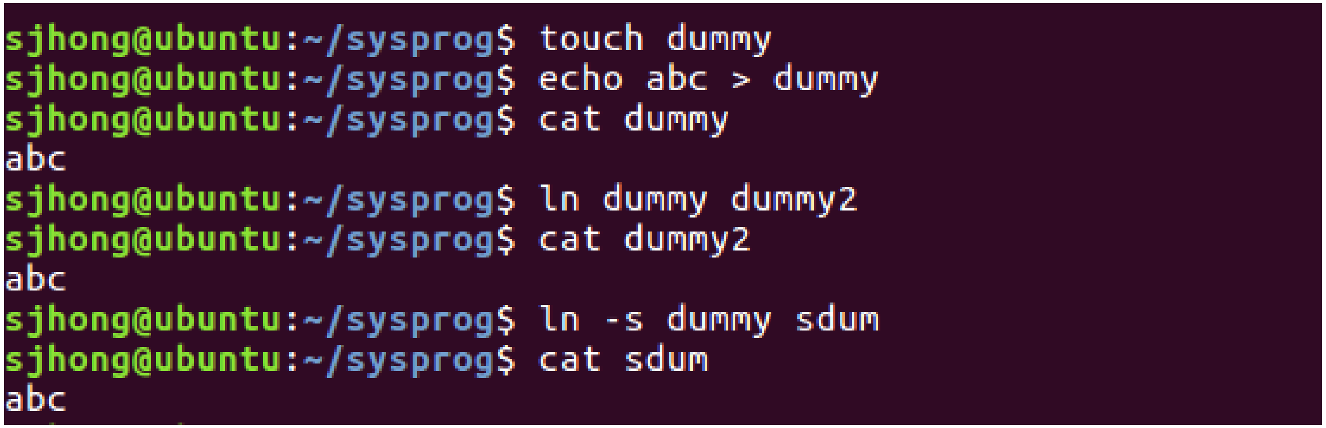
System Programming

실습

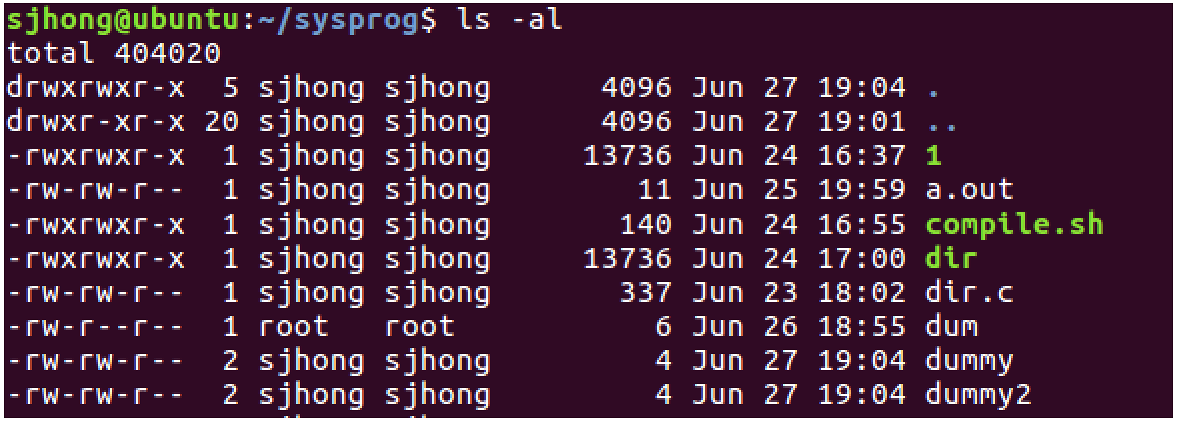
gcc 사용하기

❑ link, symlink

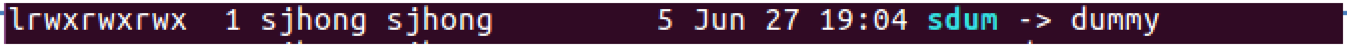
– hardlink와 softlink의 차이



```
sjhong@ubuntu:~/sysprog$ touch dummy
sjhong@ubuntu:~/sysprog$ echo abc > dummy
sjhong@ubuntu:~/sysprog$ cat dummy
abc
sjhong@ubuntu:~/sysprog$ ln dummy dummy2
sjhong@ubuntu:~/sysprog$ cat dummy2
abc
sjhong@ubuntu:~/sysprog$ ln -s dummy sdum
sjhong@ubuntu:~/sysprog$ cat sdum
abc
```



```
sjhong@ubuntu:~/sysprog$ ls -al
total 404020
drwxrwxr-x  5 sjhong sjhong      4096 Jun 27 19:04 .
drwxr-xr-x 20 sjhong sjhong      4096 Jun 27 19:01 ..
-rwxrwxr-x  1 sjhong sjhong     13736 Jun 24 16:37 1
-rw-rw-r--  1 sjhong sjhong        11 Jun 25 19:59 a.out
-rwxrwxr-x  1 sjhong sjhong       140 Jun 24 16:55 compile.sh
-rwxrwxr-x  1 sjhong sjhong     13736 Jun 24 17:00 dir
-rw-rw-r--  1 sjhong sjhong       337 Jun 23 18:02 dir.c
-rw-r--r--  1 root   root           6 Jun 26 18:55 dum
-rw-rw-r--  2 sjhong sjhong         4 Jun 27 19:04 dummy
-rw-rw-r--  2 sjhong sjhong         4 Jun 27 19:04 dummy2
```



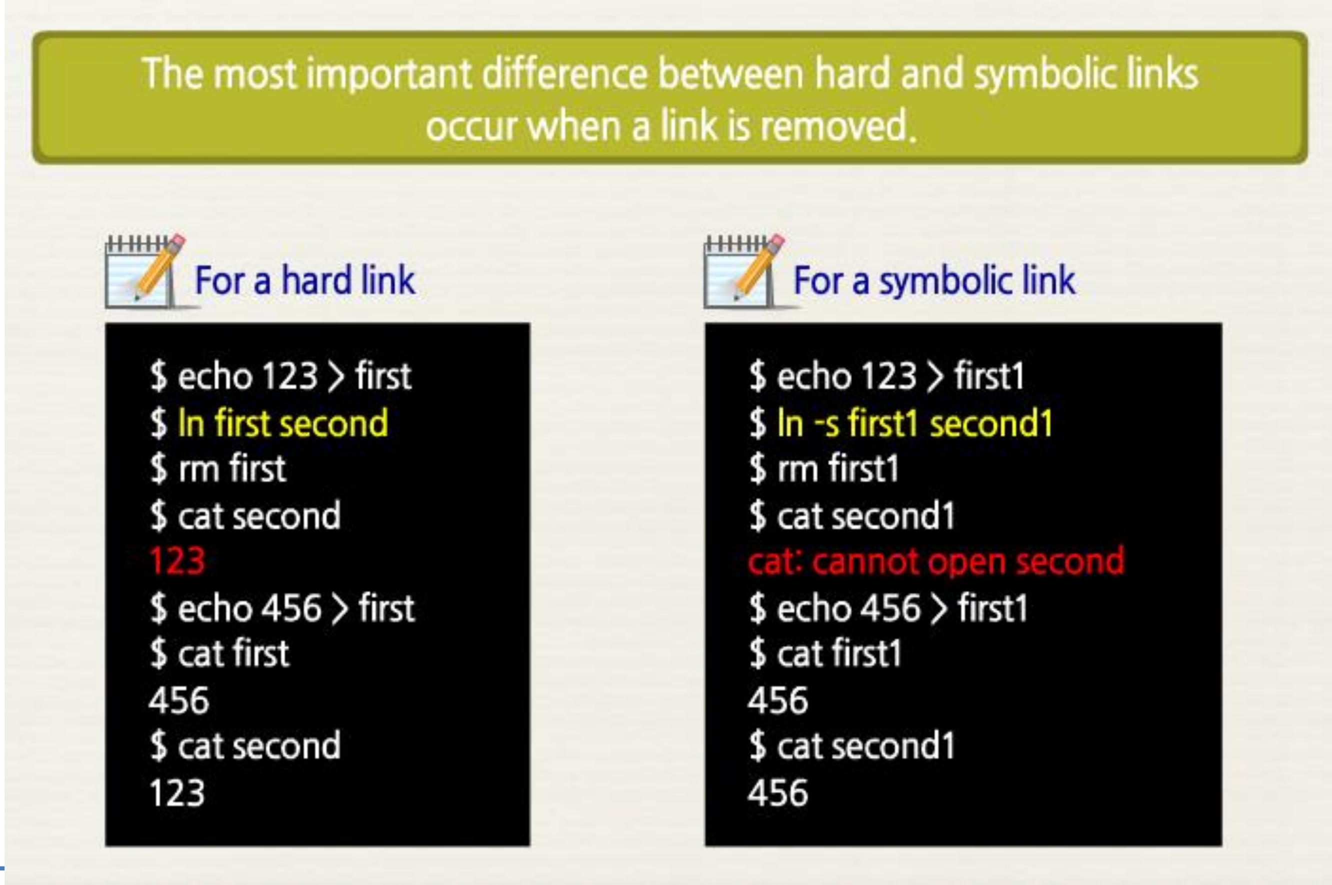
```
lrwxrwxrwx  1 sjhong sjhong         5 Jun 27 19:04 sdum -> dummy
```

실습

gcc 사용하기

❑ link, symlink

– hardlink와 softlink의 차이



The most important difference between hard and symbolic links occur when a link is removed.

For a hard link

```
$ echo 123 > first
$ ln first second
$ rm first
$ cat second
123
$ echo 456 > first
$ cat first
456
$ cat second
123
```

For a symbolic link

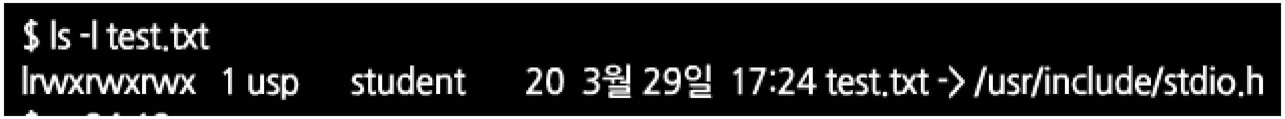
```
$ echo 123 > first1
$ ln -s first1 second1
$ rm first1
$ cat second1
cat: cannot open second
$ echo 456 > first1
$ cat first1
456
$ cat second1
456
```

실습

gcc 사용하기

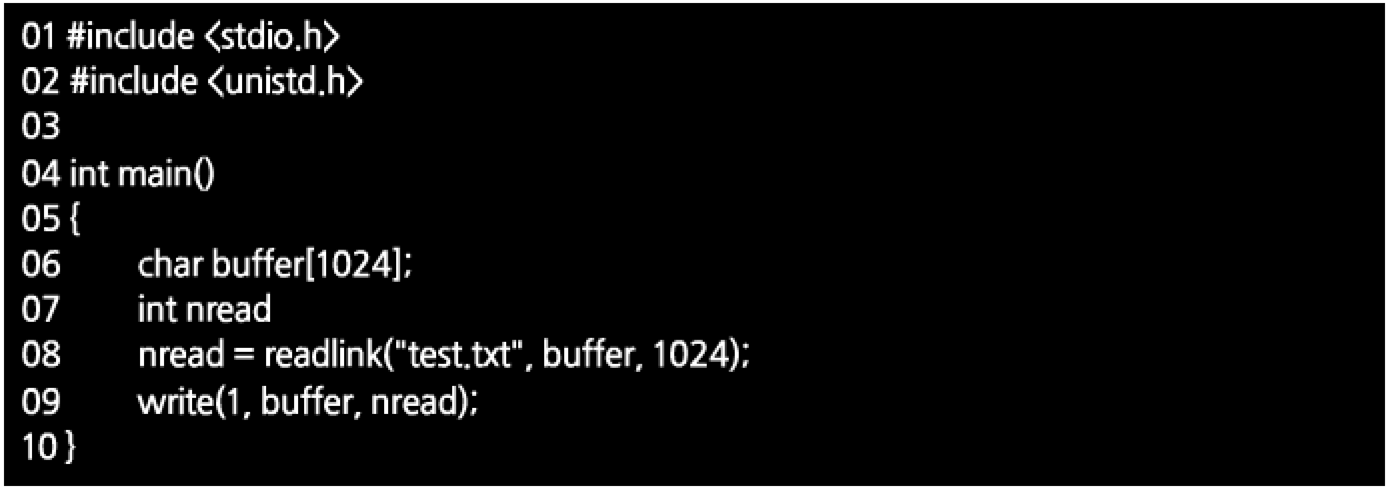
❑ Readlink

- ln –s명령으로 test.txt 링크가 아래와 같이 가리키도록 수행하고,



```
$ ls -l test.txt
lrwxrwxrwx  1 usp     student      20  3월 29일  17:24 test.txt -> /usr/include/stdio.h
```

- 아래 코드를 컴파일해서 결과 값을 확인해본다.



```
#include <stdio.h>
#include <unistd.h>

int main()
{
        char buffer[1024];
        int nread
        nread = readlink("test.txt", buffer, 1024);
        write(1, buffer, nread);
}
```

실습

gcc 사용하기

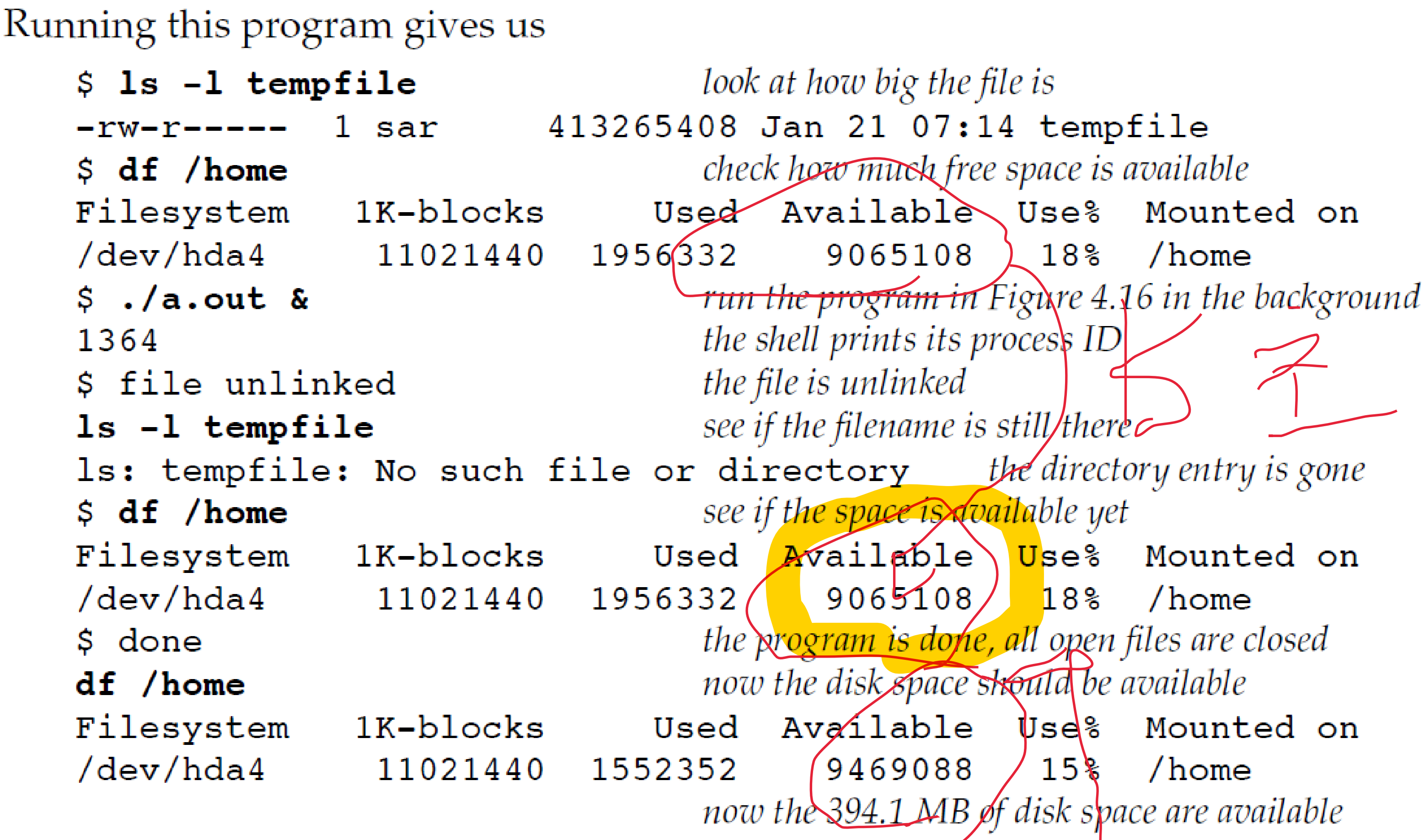
❑ 책자(이론) 예제 코드 컴파일 및 실행 해보기

- Prog. 4.16
  - 이전 실습내용을 참고로 tempfile(size : 413265408 bytes) 만들어서후 테스트해보기
- Prog. 4.21
- Prog. 4.22
  - 첫번째 인자값으로 경로를 입력해볼 것
- Prog. 4.25

실습

파일 & 디렉토리 system call

❑ Prog. 4. 16 실행



Running this program gives us

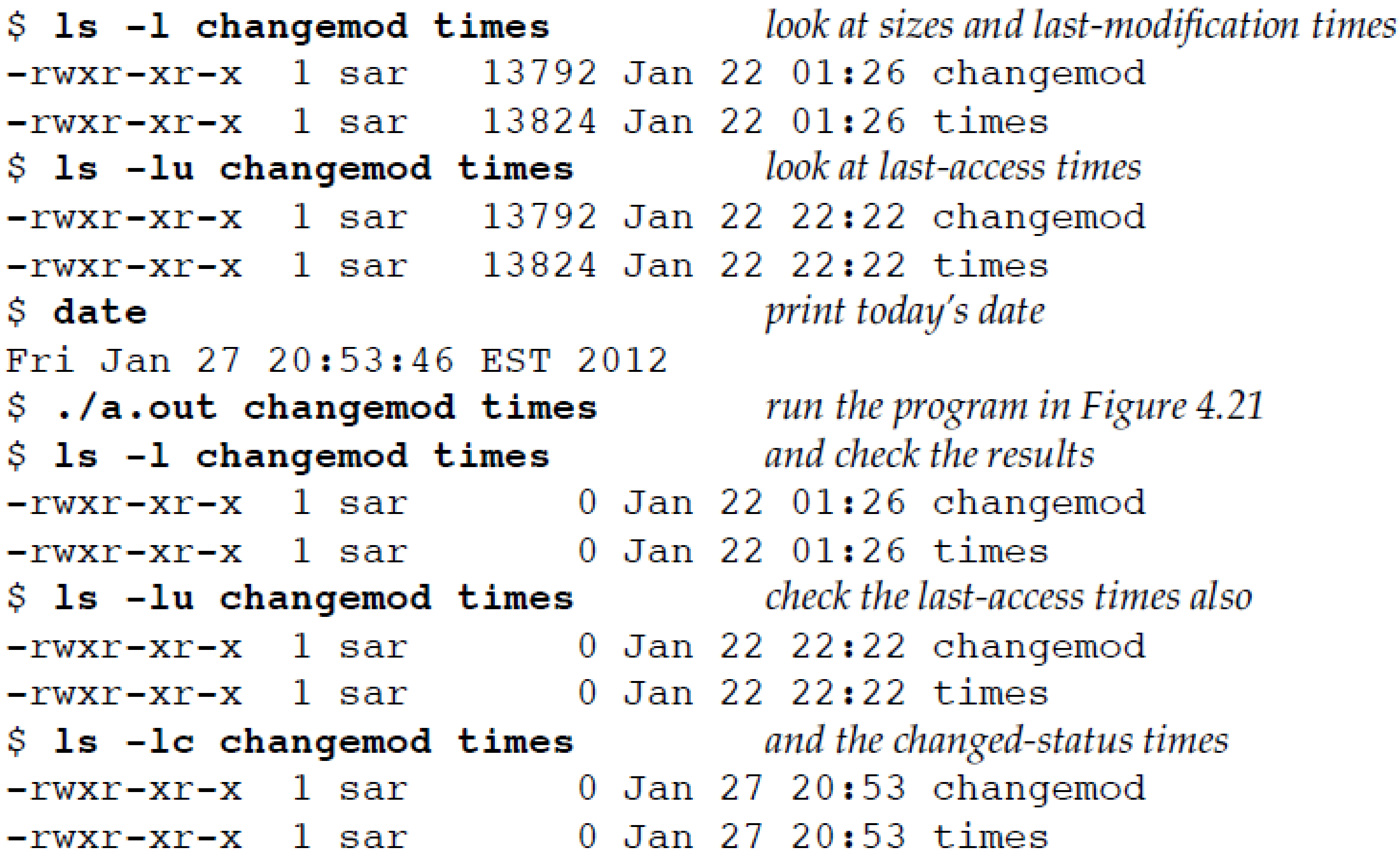
```
$ ls -l tempfile                    look at how big the file is
-rw-r-----  1 sar    413265408 Jan 21 07:14 tempfile
$ df /home                          check how much free space is available
Filesystem   1K-blocks     Used  Available  Use%  Mounted on
/dev/hda4     11021440  1956332    9065108   18%  /home
$ ./a.out &                         run the program in Figure 4.16 in the background
1364                                the shell prints its process ID
$ file unlinked                     the file is unlinked
ls -l tempfile                      see if the filename is still there
ls: tempfile: No such file or directory     the directory entry is gone
$ df /home                          see if the space is available yet
Filesystem   1K-blocks     Used  Available  Use%  Mounted on
/dev/hda4     11021440  1956332    9065108   18%  /home
$ done                              the program is done, all open files are closed
df /home                            now the disk space should be available
Filesystem   1K-blocks     Used  Available  Use%  Mounted on
/dev/hda4     11021440  1552352    9469088   15%  /home
                                    now the 394.1 MB of disk space are available
```

5조

실습

파일 & 디렉토리 system call

❑ **Prog. 4. 21 실행**

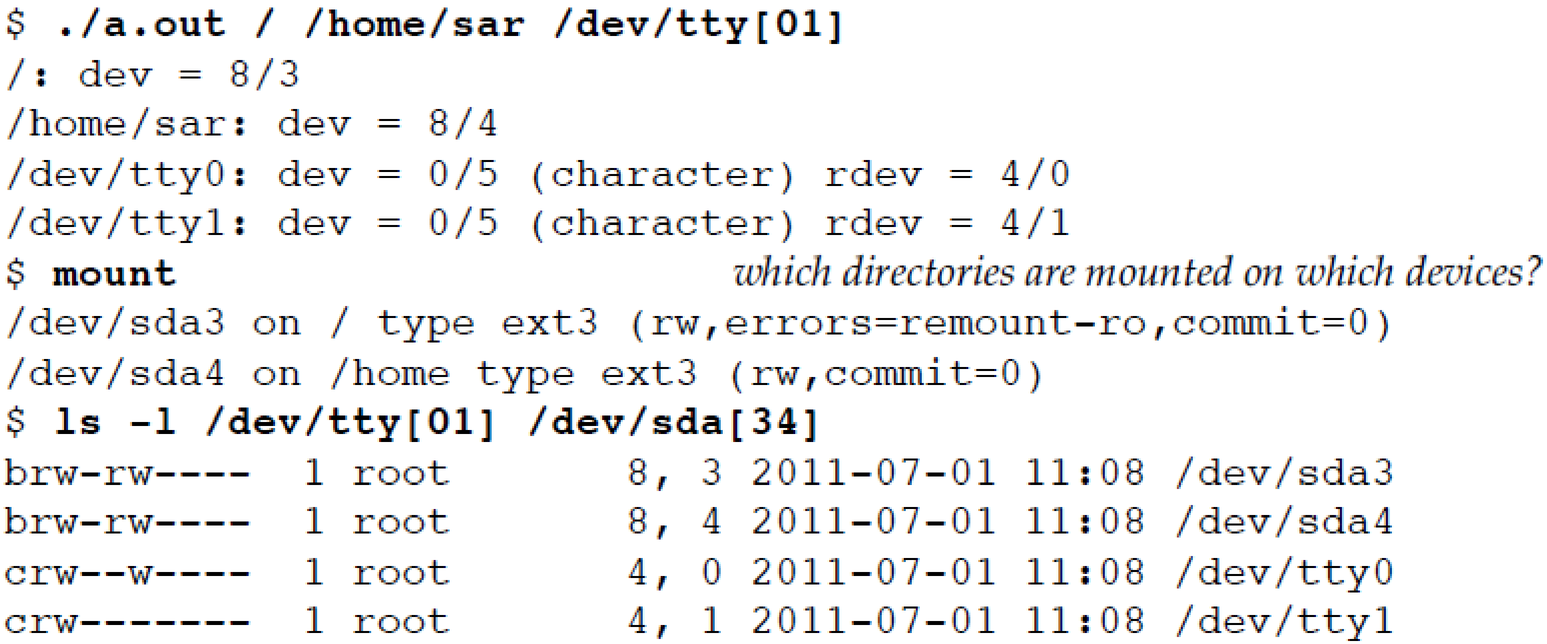


```
$ ls -l changemod times          look at sizes and last-modification times
-rwxr-xr-x  1 sar   13792 Jan 22 01:26 changemod
-rwxr-xr-x  1 sar   13824 Jan 22 01:26 times
$ ls -lu changemod times         look at last-access times
-rwxr-xr-x  1 sar   13792 Jan 22 22:22 changemod
-rwxr-xr-x  1 sar   13824 Jan 22 22:22 times
$ date                           print today's date
Fri Jan 27 20:53:46 EST 2012
$ ./a.out changemod times        run the program in Figure 4.21
$ ls -l changemod times          and check the results
-rwxr-xr-x  1 sar       0 Jan 22 01:26 changemod
-rwxr-xr-x  1 sar       0 Jan 22 01:26 times
$ ls -lu changemod times         check the last-access times also
-rwxr-xr-x  1 sar       0 Jan 22 22:22 changemod
-rwxr-xr-x  1 sar       0 Jan 22 22:22 times
$ ls -lc changemod times         and the changed-status times
-rwxr-xr-x  1 sar       0 Jan 27 20:53 changemod
-rwxr-xr-x  1 sar       0 Jan 27 20:53 times
```

실습

파일 & 디렉토리 system call

❑ Prog. 4. 25 실행



```
$ ./a.out / /home/sar /dev/tty[01]
/: dev = 8/3
/home/sar: dev = 8/4
/dev/tty0: dev = 0/5 (character) rdev = 4/0
/dev/tty1: dev = 0/5 (character) rdev = 4/1
$ mount
/dev/sda3 on / type ext3 (rw,errors=remount-ro,commit=0)
/dev/sda4 on /home type ext3 (rw,commit=0)
$ ls -l /dev/tty[01] /dev/sda[34]
brw-rw----  1 root       8, 3 2011-07-01 11:08 /dev/sda3
brw-rw----  1 root       8, 4 2011-07-01 11:08 /dev/sda4
crw--w----  1 root       4, 0 2011-07-01 11:08 /dev/tty0
crw-------  1 root       4, 1 2011-07-01 11:08 /dev/tty1
```

which directories are mounted on which devices?

Thank you for your attention !!

Q and A