오픈소스 소프트웨어 실습

Open-Source Software Lab

#4

실습 담당 조교 연락처

◆실습 조교 : 김민곤

◆연구실 : 학연산클러스터 601호

◆이메일 : phenix235@hanyang.ac.kr

생성파일 출력 : touch

$ touch 파일

파일 크기가 0인 이름만 있는 빈 파일을 만들어 준다.

```
[1111222333@node1 class]$ touch cs2.txt
[1111222333@node1 class]$ ls -asl
합계 8
0 drwxrwxr-x.  5 1111222333 1111222333   72  3월  28 03:25 .
4 drwx------. 20 1111222333 1111222333 4096  3월  28 03:24 ..
4 -rw-rw-r--.  1 1111222333 1111222333   12  3월  20 12:13 cs1.txt
0 -rw-rw-r--.  1 1111222333 1111222333    0  3월  28 03:25 cs2.txt
```

페이지 단위로 파일 내용 보기 : more

$ more 파일

파일(들)의 내용을 페이지 단위로 화면에 출력한다.

```
[111222333@node1 class]$ vi cs2.txt
```

```
[111222333@node1 class]$ more cs2.txt
```

페이지 단위로 파일 내용 보기 : more

```
Unix is a multitasking, multi-user computer operating system originally
developed in 1969 by a group of AT&T employees at Bell Labs, including
Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy,
and Joe Ossanna.

The Unix operating system was first developed in assembly language,
but by 1973 had been almost entirely recoded in C, greatly facilitating
its further development and porting to other hardware.
Today's Unix system evolution is split into various branches,
developed over time by AT&T as well as various commercial vendors,
universities (such as University of California, Berkeley's BSD),
and non-profit organizations.

The Open Group, an industry standards consortium, owns the UNIX trademark.
Only systems fully compliant with and certified according to the Single
UNIX Specification are qualified to use the trademark;
others might be called Unix system-like or Unix-like,
although the Open Group disapproves[1] of this term.
However, the term Unix is often used informally to denote any operating
system that closely resembles the trademarked system.

During the late 1970s and early 1980s, the influence of Unix in academic
circles led to large-scale adoption of Unix(particularly of the BSD variant,
--More--(59%)
```

파일 앞부분 보기 : head

$ head [-n] 파일

파일(들)의 앞부분을 화면에 출력한다. 파일을 지정하지 않으면 표준입력 내용을 대상으로 한다.

```
[1111222333@node1 class]$ head cs2.txt
```

파일 앞부분 보기 : head

```
Linux, which is used to power data centers, desktops, mobile phones,
and embedded devices such as routers, set-top boxes or e-book readers.
Today, in addition to certified Unix systems such as those already
mentioned, Unix-like operating systems such as MINIX, Linux, Android,
and BSD descendants (FreeBSD, NetBSD, OpenBSD, and DragonFly BSD) are
commonly encountered.

The term traditional Unix may be used to describe a Unix or
an operating system that has the characteristics of either Version 7
Unix or UNIX System V.

[1111222333@node1 class]$
[1111222333@node1 class]$ head cs2.txt
Unix is a multitasking, multi-user computer operating system originally
developed in 1969 by a group of AT&T employees at Bell Labs, including
Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy,
and Joe Ossanna.

The Unix operating system was first developed in assembly language,
but by 1973 had been almost entirely recoded in C, greatly facilitating
its further development and porting to other hardware.
Today's Unix system evolution is split into various branches,
developed over time by AT&T as well as various commercial vendors,
[1111222333@node1 class]$
```

파일 뒷부분 보기 : tail

$ tail [-n] 파일

파일(들)의 뒷부분을 화면에 출력한다. 파일을 지정하지 않으면 표준 입력 내용을 대상으로 한다.

```
[1111222333@node1 class]$ tail cs2.txt
and embedded devices such as routers, set-top boxes or e-book readers.
Today, in addition to certified Unix systems such as those already
mentioned, Unix-like operating systems such as MINIX, Linux, Android,
and BSD descendants (FreeBSD, NetBSD, OpenBSD, and DragonFly BSD) are
commonly encountered.

The term traditional Unix may be used to describe a Unix or
an operating system that has the characteristics of either Version 7
Unix or UNIX System V.

[1111222333@node1 class]$
```

파일 복사 : cp(copy)

$ cp [-i] 파일1 파일2

파일 1을 파일 2에 복사한다. –i는 대화형 옵션이다

```
[1111222333@node1 class]$ ls
cs1.txt   cs2.txt   dest   temp   test
[1111222333@node1 class]$ cp cs2.txt cs3.txt
[1111222333@node1 class]$ ls
cs1.txt   cs2.txt   cs3.txt   dest   temp   test
[1111222333@node1 class]$ vi cs3.txt
```

파일 복사 : cp(copy)

$ cp [-i] 파일1 파일2

파일 1을 파일 2에 복사한다. –i는 대화형 옵션이다

- 대화형 옵션: cp –i
  - ✓ 복사 대상 파일과 이름이 같은 파일이 이미 존재하면 덮어쓰기(overwrite)
  - ✓ 보다 안전한 사용법: 대화형 –i(interactive) 옵션을 사용

```
[1111222333@node1 class]$ cp -i cs1.txt cs3.txt
cp: overwrite `cs3.txt'? n
[1111222333@node1 class]$
```

파일 이동 : mv(move)

$ mv [-i] 파일1 파일2

파일 1의 이름을 파일 2로 변경한다. –i는 대화형 옵션이다

```
[1111222333@node1 class]$ mv cs2.txt cs4.txt
[1111222333@node1 class]$ ls -l
합 계  12
-rw-rw-r--. 1 1111222333 1111222333   12  3월  28 04:03 cs1.txt
-rw-rw-r--. 1 1111222333 1111222333 2089  3월  28 03:52 cs3.txt
-rw-rw-r--. 1 1111222333 1111222333 2089  3월  28 03:51 cs4.txt
```

파일 삭제 : rm(remove)

$ rm [-i] 파일

파일(들)을 삭제한다. –i는 대화형 옵션이다.

```
[1111222333@node1 class]$ rm cs4.txt
[1111222333@node1 class]$ ls -l
합 계  8
-rw-rw-r--. 1 1111222333 [1111222333@node1 class]$ rm test
rm: cannot remove `test': 디 렉 터 리 입 니 다 ) 04:03 cs1.txt
-rw-rw-r--. 1 1111222333 1111222333 2089  3월  28 03:52 cs3.txt
drwxrwxr-x. 3 1111222333 1111222333   18  3월  20 13:06 dest
drwxrwxr-x. 2 1111222333 1111222333    6  3월  20 13:01 temp
drwxrwxr-x. 2 1111222333 1111222333    6  3월  20 13:01 test
```

```
[1111222333@node1 class]$ rm -i cs1.txt
rm: remove 일 반  파 일  `cs1.txt'? n
[1111222333@node1 class]$
```

```
[1111222333@node1 class]$ rm test
rm: cannot remove `test': 디 렉 터 리 입 니 다
```

링크 : ln(link)

$ ln [-s] 파일1 파일2

파일 1에 대한 새로운 이름(링크)로 파일 2를 만들어 준다.

-s 옵션은 심볼릭 링크

$ ln [-s] 파일1 디렉터리

파일 1에 대한 링크를 지정된 디렉터리에 같은 이름으로 만들어 준다.

```
[1111222333@node1 class]$ ln cs3.txt cs2.txt
[1111222333@node1 class]$ ls -l
합계 12
-rw-rw-r--. 1 1111222333 1111222333   12  3월  28 04:03 cs1.txt
-rw-rw-r--. 2 1111222333 1111222333 2089  3월  28 03:52 cs2.txt
-rw-rw-r--. 2 1111222333 1111222333 2089  3월  28 03:52 cs3.txt
drwxrwxr-x. 3 1111222333 1111222333   18  3월  20 13:06 dest
drwxrwxr-x. 2 1111222333 1111222333    6  3월  20 13:01 temp
```

링크 : ln(link)

- 심볼릭 링크

```
[1111222333@node1 class]$ ln -s cs3.txt cs4.txt
[1111222333@node1 class]$ ls -l
합계 12
-rw-rw-r--. 1 1111222333 1111222333   12  3월  28 04:03 cs1.txt
-rw-rw-r--. 2 1111222333 1111222333 2089  3월  28 03:52 cs2.txt
-rw-rw-r--. 2 1111222333 1111222333 2089  3월  28 03:52 cs3.txt
lrwxrwxrwx. 1 1111222333 1111222333    7  3월  28 04:34 cs4.txt -> cs3.txt
drwxrwxr-x. 3 1111222333 1111222333   18  3월  20 13:06 dest
drwxrwxr-x. 2 1111222333 1111222333    6  3월  20 13:01 temp
```

링크 : ln(link)

- 다른 디렉터리로 링크

```
[lll1222333@node1 class]$ mkdir test
[lll1222333@node1 class]$ ls
cs1.txt  cs2.txt  cs3.txt  cs4.txt  dest  temp  test
```

```
[lll1222333@node1 class]$ ln cs2.txt test
```

```
[lll1222333@node1 class]$ ls -R
.:
cs1.txt  cs2.txt  cs3.txt  cs4.txt  dest  temp  test

./dest:
dir1

./dest/dir1:

./temp:

./test:
cs2.txt
```

링크 : ln(link)

- 다른 디렉터리로 링크

```
[1111222333@node1 class]$ ln -s cs3.txt test
[1111222333@node1 class]$ ls -R
.:
cs1.txt  cs2.txt  cs3.txt  cs4.txt  dest  temp  test

./dest:
dir1

./dest/dir1:

./temp:

./test:
cs2.txt  cs3.txt
```

링크 : ln(link)

- 다른 디렉터리로 링크

```
[1111222333@node1 class]$ cd test
[1111222333@node1 test]$ ls -l
합 계  4
-rw-rw-r--. 3 1111222333 1111222333 2089  3월  28 04:39 cs2.txt
lrwxrwxrwx. 1 1111222333 1111222333    7  3월  28 04:41 cs3.txt -> cs3.txt
```

쉘(Shell)

- 쉘의 역할
  - ✓ 쉘은 사용자와 운영체제 사이에 창구 역할을 하는 소프트웨어
  - ✓ 명령어 처리기(command processor)

```
[1111222333@node1 test]$ echo $SHELL
/bin/bash
```

끝