1.쉘이란 무엇인인가?

쉘이란 키보드로 입력한 명령어를 운영체제에 전달하여 이 명령어를 실행하게 하는 프로그램

대부분의 리눅스 배포판은 bash라고 하는 GNU 프로젝트의 쉘 프로그램을 제공한다.

1.1 터미널 에뮬레이터

GUI 환경에서 쉘과 직접 작업할 수 있도록 도와주는 프로그램

1.2 첫번째 키입력

leedo-com@leedo-Pc:~$

쉘 프롬프트라 부르며 쉘이 입력 가능상태(보통은 username@machinename 같은 형식 뒤이어 현재 작업 디렉토리와 달러표시

만약 마지막 글자가 달려($)가 아니라 해쉬 표시(#)라면 현재 터미널 세션이 슈퍼유저 권한을 가졌다는 뜻, 즉 루트(root)사용자로 로그인 했거나 관리자 권한을 가진 터미널 에뮬레이터를 사용

1.2.1 명령어 히스토리

방금 입력한 명령어를 다시 보려면 위 아래 방향키를 통해 볼수 있음. 대부분의 리눅스 배포판들은 가장 최근 500개의 명령을 기억할 수 있 다.(Leedo:::ubuntu 14버전은 더기억하는듯...)

leedo-com@leedo-Pc:~$ history <=Leedo:::이 명령어를 통해서도 가능

1.2.2 커서 이동

커서를 명령어 어느 부분이라도 이동할 수 있어 명령어를 쉽게 편집할 수 있다.

1.3 간단한 명령어 실행하기

 Leedo:::현재 시간과 날짜를 표시

date ::: 현재 시간과 날짜를 표시

cal ::: 현재 날짜의 달력 표시

df ::: 현재 사용중인 디스크 정보와 사용 가능한 디스크의 용량

free ::: 메모리 사용 현황 정보

1.4 터미널 세션 종료

터미녈 세션을 종료하는 방법은 두가지 이다.

1. 직접 터미널 에뮬레이터 창을 닫는다.

2. 쉘 프로프트에 exit 명령어를 입력한다.

BONUS 보이지 않는 터미널(에뮬레이터)

사실 실행중인 터미널이 없는 것 처럼 보여도 다수의 터미널 세션들이 그래픽 환경에 가려 보이지 않은 뿐 끊임없이 실행되고 있다. 이런 것들을 가상 터미널 또는 가상 콘솔이라고 하는데, 대부분의 리눅스 배포판에서 ctrl-alt-F1 ~F6 단축키로 시스템상에 있는 가상 터미널에 접근할 수 있다. 세션에 연결하면 사용자 이름과 비밀번호를 입력할 수 있도록 로그인 프롬프트가 표시된다. 다른 가상 콘솔로 이동하려면 alt-F1 ~ F6 사이의 키를 입력하고, 그래픽 환경으로 복귀하려면 alt-F7 키를 사용하면 된다.

---------------------------------------------------------------------------------------------------------------------------

2.파일시스템 탐색

이 장에서는 다음 명령어 들을 소개한다.

pwd - 현재 작업 디렉토리를 표시

cd - 디렉토리 변경

ls - 디렉토리 내용 나열

2.1 파일 시스템 트리 구조의 이해

계층적인 디렉토리 구조로 파일을 구성하며, 최상의 디렉토리를 루트(root) 디렉토리라고 한다.

윈도우즈, 리눅스와 같이 유닉스형 시스템의 차이점은 윈도우즈는 저장장치마다 개별 파일 시스템으로 관리하는 반면 유닉스시스템에서는 아무리 많은 저장장치가 설치되었다 해도 단일 파일 시스템으로 관리한다는 점이다. 유닉스형 시스템의 저장장치들은 시스템 유지보수를 담당하는 시스템 관리자의 재량에 따라 다양한 위치에 설치(정확한 표현은 장착 또는 마운트)된다.

2.2 현재 작업 디렉토리

pwd (print Working Directory)



시스템에 처음 로그인하면 홈 디렉토리가 현재 작업 디렉토리가 된다.

사용자 계정마다 고유의 홈 디렉토르를 갖게 되는데, 일반 사용자로 시스템을 사용할때 파일 쓰기 권한이 부여되는 유일한 공간

2.3 디렉토리 목록 표시

ls : 현재 작업 디렉토리에 파일과 하위 디렉토리 표시



자세한건 3장에서 다룸

2.4 현재 작업 디렉토리변경

cd(change directory) : 현재 작업중인 디렉토리를 변경

2.4.1 절대경로명

루트 디렉토리에서 원하는 디렉토리 또는 파일까지의 경로에 대해여 각 디렉토리 위치들 하나하나를 명시



2.4.2 상대경로명

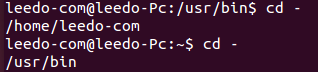
현재 디렉토리를 시작점으로 하며, 상대경로명을 표시하려면 파일시스템 트리상의 상대적인 위치를 대신 표현해주는 특수 기호를 사용해야 하는데 바로 .(점 : 현재 작업 디렉토리)과 ..(점점 : 상위 디렉토리)

2.4.3 유용한 단축 표현들

cd : 작업디렉토리를 홈 디렉토리로 변경

cd - : 작업디렉토리를 이전 작업 디렉토리로 변경

cd ~username : username의 홈 디렉토리로 작업 디렉토리를 변경



BONUS 파일명에 관한 중요한 몇가지 사실

1 .(마침표)로 시작하는 파일명을 가진 파일들은 보이지 않는다.

2 유닉스처럼 리눅스도 파일명과 명령어는 대소문자를 구별한다.

3 다른운영체제들과는 달리 "파일확장자" 개념이 없기 때문에 원하는대로 파일명을 만들수 있다.

4 리눅스에서는 띄어쓰기와 구두점 기호가 포함된 긴 파일명이 허용된다. 하지만 파일명에 구두점 기호를 사용할때, 마침ㅂ표, 대시 및 밑줄표시만이 사용가능하다. 무엇보다 가장중요한것은 파일명에 공백(스페이스)를 포함하지 말라는 것인다. 파일며에 공백이 있으면 커맨드라인에서 입력이 번거롭다.

---------------------------------------------------------------------------------------------------------------------------

3. 시스템 살펴보기

시작하기에 앞서 알아두어야 할 명령어

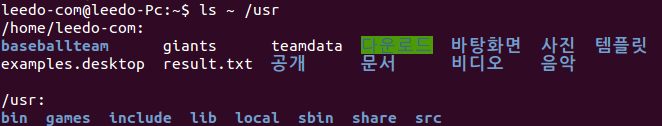
ls - 디렉토리 내용 나열

file - 파일타입 확인

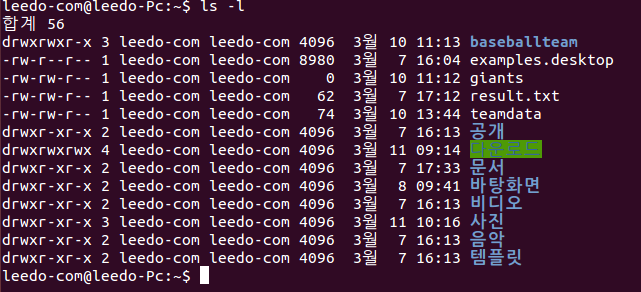
less - 파일 내용표시

3.1 재미있는 ls 명령어

한번에 여러 디렉토리 목록을 보수 있다. 예) 홈 디렉토리와 / usr 디렉토리



자세한 속성까지 확인하려면 ls -l (leedo:: ll 과도 같음)



3.1.1 명령어 옵션과 명령인자

대부분의 명령어를 입력하고 그다음에 -(대시)와 함계 옵션을 명시한다.

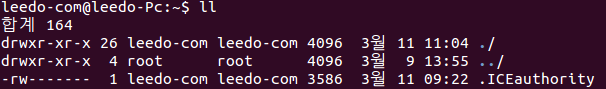
-l 과 같이 단축옵션을 사용하기도 하지만 GNU 프로젝트에서 사용하는 명령어들을 포함하여 많은 명령어들이 long 옵션을 제공한다.

long 옵션에는 -- 기호를 선행하여 쓴다.

주로 많이 사용되는 ls 옵션

|  |  |  |
| --- | --- | --- |
| 옵션 | long 옵션 | 설명 |
| -a | --all | 모든 파일보기, 숨긴 파일 포함 |
| -d | --directory | 디렉토리 내용이 아닌 디렉토리 자체 정보를 자세히 확인가능 |
| -F | --classify | 지시 문자를 추가, 예를들면, 디렉토리명이 끝에 / 를 덧붙인다. |
| -h | --human-readable | -l 옵션과 함께 사용하여 파일 크기를 사람이 인식하기 쉽게 표시 |
| -l |  | 좀 더 자세한 정보 출력 |
| -r | --reverse | 출력 결과를 역순으로 표시 |
| -s |  | 파일 크기순으로 정렬 |
| -t |  | 파일 수정 시간순으로 정렬 |

3.1.2 long 포맷으로 출력 결과보기



-rw------- : 파일 접근권한

1 : 하드 링크의 수(링크는 뒤에 설명)

leedo-com : 파일 소유자

leedo-com : 파일 소유 그룹

3586 : 파일크기

3월 11일 09:22 : 파일 마지막 수정 날짜와 시간

.ICEauthority : 파일명

3.2 file 명령어로 파일 타입 확인



사실 리눅스와 같은 유닉스형 운영체제에서는 "모두 다 파일이다" 라는 기본적인 개념이 있다

3.3 less 명령어로 파일정보보기

less 명령어는 텍스트 파일을 볼 때 사용하는 프로그램이다. 프로그램을 실행하면 스크롤하여 확인할 수 있다.



명령키

|  |  |
| --- | --- |
| 명령키  ----------------------------------------------------------  PAGE UP 또는 B  PAGE DOWN 또는 SPACE BAR  위쪽 방향키  아래쪽 방향키  G  1G 또는 g  /문자열  n  h  q | 실행  ----------------------------------------------------------  한 페이지 위로  한 페이지 아래로  한 줄 위로  한 줄 아래로  텍스트 파일 맨 마지막으로 이동  택스트 파일 맨 처음으로 이동  아래방향으로 진행되며 입력된 문자열 찾기  이전 검색어의 다음 찾기  도움말 보기  프로그램 종료 |

BONUS : “Less Is More “ 단순할 수록 풍부해진다.

3.4 함께 탐험해 보자

리눅스의 파일시스템 배치 형식은 리눅스 파일시스템 계층 표증에 따라 설계 됨

리눅스 시스템 디렉토리

/ ::: 루트

/dev ::: 주변 장치들이 파일로 등록

/etc ::: 시스템 설정 파일이 들어 있는 디레토리, 사용자, 그룹정보, 파일시스템 테이블 같은....

/bin ::: 사용자가 사용하는 가장 기본적인 명령들이 실행파일 형태로 저장

/lib ::: 공유 라이브러리 파일들이 저장 되어있는 디렉토리

/home ::: 사용자 계정을 만들면 사용자 계정의 이름과 동일한 홈 디렉토리가 /home 아래 만들어짐

/root ::: 루트계정을 위해 제공되는 홈 디렉토리

/proc ::: 시스템 정보를 제공하는 가상 파일 시스템 디렉토리, CPU 정보, 인터럽트 목록, 입출력주소 목록과 같은 시스템 상태를 모니터링할 수 있다.

/sbin ::: 시스템 관리 명령들이 들어있는 디렉토리

/tmp ::: 임시로 파일을 생성 또는 삭제하는 공간입니다. 이 디렉토리에 저장한 파일은 언제 삭제될지 보장할 수 없다.

/var ::: 시스템을 운영하면서 생기는 각종 임시파일(시스템로그, 스풀, 전자메일)을 저장하는 디렉토리, 크기가 계속해서 변하는 임시 파일을 저장

/usr ::: 시스템, 응용 프로그램에서 필요한 파일들이 저장되어 있는 디렉토리

3.5 심볼릭 링크

디렉토리 정보의 첫글자가 l이고 파일명이 두개인 특수한 종류의 파일이 심볼릭 링크다.

대부분의 유닉스형 시스템에서는 하나의 파일에 여러 이름을 부여할 수 있다.

---------------------------------------------------------------------------------------------------------------------------

4 파일과 디렉토리 조작

이 장에서 설명하는 명령어

cp - 파일 및 디렉토리 복사하기

mv - 파일 및 디렉토리 이동 그리고 이름바꾸기

mkdir - 디렉토리 새로 만들기

rm - 파일 및 디렉토리 삭제하기

ln - 하드링크 또는 심볼릭 링크 만들기

4.1 와일드 카드

쉘은 굉장히 많은 파일명을 사용하기 때문에 간단하게 파일명의 그룹을 지정할 수 있도록 특수한 문자들은 지원한다. 이러한 특수 문자들을 와일드카드라고 한다.

와일드카드

|  |  |
| --- | --- |
| 와일드카드 | 매칭문자 |
| \*  ?  [characters]  [!characters]  [[:class:]] | 모든 문자  모든 하나의 문자  characters 문자셋에 포함된 문자  characters 문자셋에 포함되지 않은 문자  지정된 문자 클래스에 포함된 문자 |

가장 많이 사용되는 문자 클래스

|  |  |
| --- | --- |
| 문자클래스 | 매칭 문자 |
| [:almun:]  [:alpha:]  [:digit:]  [:lower:]  [:upper:] | 모든 알파벳과 숫자 문자  모든 알파벳 문자  모든 숫자  모든 소문자  모든 대문자 |

와일드 카드 사용 예시

|  |  |
| --- | --- |
| [abc]\*  \*[[:lower:]123] | a,b,c 로 시작하는 모든 파일일  파일명이 소문자로 큰나거나 1,2,3으로 끝나는 파일 |

4.2 mkdir - 디렉토리 생성

mkdir directory….

ex) mkdir dir1 dir2 dir3

4.3 cp - 파일 및 디렉토리 복사

cp item1 item2

cp option

|  |  |  |
| --- | --- | --- |
| 옵션(-) | 옵션(--) | 설명 |
| -a  -i  -r  -u  -v | --archive  --interactive  --recursive  --update  --verbose | 파일 및 디렉토리뿐 아니라 소유자 및 권한 정보같은 속성까지 모드 복사. 반면 일반적으로는 복사를 하는 사용자의 기본적인 속성 복사  기존파일을 덮어쓰기 전헤 확인 메세지를 보여주는 옵션  디렉토리와 그안의 내용까지 복사할때  어떤 디렉토리에 있는 파일을 다른 디렉토리로 복사할때, 그 디렉토리에는 없거나 최신 버전인 파일만을 복사하기 위해서 이옵션을 사용한다.  복사가 완료 되었다는 메세지를 보여주는 옵션이다. |

cp use example

|  |  |
| --- | --- |
| 사용예제 | 결과 |
| cp file1 file2 | file1을 file2로 복사하기, file2라는 파일이 이미 있다면 file1 내용을 그대로 덮어쓰게 된다. file2가 없으면 새로 생성 |
| cp -i file1 file2 | 위와 같이 복사가 이루어지지만 file2 파일이 있다면 덮어쓰기 여부에 대한 확인 메세지를 볼수 있다. |
| cp file1 file2 dir1 | file1과 file2를 dir1이라고하는 디렉토리로 복사한다. 단, dir1 디렉토리는 미리 존재해야한다. |
| cp dir1/\* dir2 | 와일드카드와 함께 사용해서 dir1 디렉토리에 있는 모든 파일을 dir2 디렉토리로 복사할 수 있다. 단, dir2디렉토리는 미리 존재해야한다. |
| cp -r dir1 dir2 | dir1 디렉토리와 그안에 있는 모든 내용을 dir2 디렉토리로 복사한다. dir2가 없으면 새로 생성될 것이고 dir1 디렉토리에 있는 모든 내용들이 복사된다. |

4.4 mv - 파일 이동과 이름변경

mv option

|  |  |
| --- | --- |
| 옵션 | 설명 |
| -i, --interactive | 기존 파일을 덮어쓰기 전에 확인 메세지를 보여주는 옵션 |
| -u, --update | 파일을 다른 디렉토리로 이동하려고 할때, 그 디렉토리에는 없거나 또는 최신버전인 파일 만을 이동하기 위해서 이 옵션을 사용한다. |
| -v, --verbose | 이동이 완료 되었다는 메세지를 보여주는 옵션이다. |

mv use example

|  |  |
| --- | --- |
| 사용예제 | 결과 |
| mv file1 file2 | file1을 file2로 이동한다. file2라는 파일이 이미 있다면 file1 내용을 그대로 덮어쓰게 된다. file2가 없으면 새로 생성, 두경우 모두 file1은 사라진다. |
| mv -i file1 file2 | 위와 같이 이루어지지만 file2 파일이 있다면 덮어쓰기 여부에 대한 확인 메세지를 볼수 있다. |
| mv file1 file2 dir1 | file1과 file2를 dir1이라고하는 디렉토리로 복사한다. 단, dir1 디렉토리는 미리 존재해야한다. |
| mv dir1 dir2 | dir1 과 그 내용까지 모두 dir2 디렉토리로 이동한다. dir2디렉토리가 기존에 없는 것이라면 새로 생성이 되고 dir1의 내용이 모두 그대로 이동한다. 그리고 dir1디렉토리는 삭제된다. |

4.5 rm - 파일 및 디렉토리 삭제

BONUS RM 사용시 주의할 점

리눅스처럼 유닉스형 운영체제는 삭제를 취소할 수 있는 멸령어가 없다. rm으로 일단 삭제가 되면 되찾을 수 없다. 리눅스는 사용자가 똑똑해서 현재 삭제하고 있음을 충분히 인지하고 있다고 가정하기 때문이다.

특히 와일드 카드는 주의

ex) rm \*.html ⇒ rm \* .html

앞처럼 써야할 것은 뒤처럼 사용하면

디렉토리에 있는 모든 파일을 삭제하고 .html 이라는 파일은 없다는 메세지를 보여줄 것이다.

TIP : rm 사용시 먼저 rm 자리에 ls 로 변경후 삭제할 파일을 확인후 rm 으로 실행한다.

rm option

|  |  |
| --- | --- |
| 옵션 | 설명 |
| -i , --interactive | 기존파일을 삭제하기 전에 확인 메세지를 보여주는 옵션 |
| -r, -- revursive | 재귀적으로 디렉토리를 삭제한다. 즉 삭제된 디렉토리에 하위 디렉토리들이 있다 해도 모두 삭제된다는 뜻이다. 디렉토리를 완전히 삭제 하려면 이 옵션을 반드시 사용해야한다. |
| -f, --force | 존재하지 않는 파일은 확인 메세지 없이 무시하라는 옵션. 이 옵션은 --interactive 옵션을 완전히 무시해보린다. |
| -v, --verbose | 삭제가 완료되었다는 메세지를 보여주는 옵션 |

rm use example

|  |  |
| --- | --- |
| 사용예제 | 결과 |
| rm file1 | file1을 완전히 삭제한다. |
| rm -i file1 | file1 파일을 삭제하기 전에 사용자 확인 메세지를 보여준다. |
| rm -r file1 dir1 | file1과 dir1 디렉토리 및 하위 내용가지 모두 삭제한다. |
| rm -rf dir1 dir2 | 위와 같이 file1과 dir1 모두 다 삭제하되 file1이나 dir1 이 존재하지 않더라도 rm 명령어가 실행된다. |

4.6 In - 링크생성(leedo ::: 아직 링크가 어떤 개념인지 모르겠음....)

하드 링크와 심볼릭 링크를 만들 때 사용한다.

4.6.1 하드링크

ln file link

기본적으로 하나의 파일에는 하나의 하드링크가 있는데 그것이 바로 파일에 이름을 만들어주는 것이다. 하드링크가 만들어질 때 그파일에 대한 디렉토리가 곧바로 생성된다.

but, 치명적인 두가지 약점

1.파일시스템 외부에 있는 파일을 참조할 수 없다.

2.하드링크는 디렉토리를 참조할 수 없다.

하드링크가 삭제될 때, 링코드 함께 사라지지만 파일 내용은 그파일의 모든 링크가 삭제될 때까지 계속 남아잇는다.

4.6.2 심볼릭 링크

ln -s item link

심볼릭 링크는 하드링크의 한계를 극복하기 위해서 탄생되었다.

심볼릭 링크는 참조될 파일이나 디렉토리를 가리키는 텍스트 포인터가 포함된 특수한 파일을 생성한다. 이러한 점에서 윈도우즈의 바로가기와 매우 흡사한 방식이다.

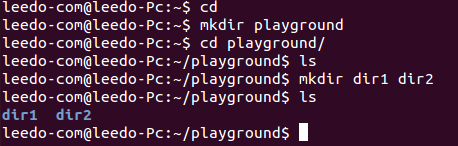
심볼릭 링크가 참조하고 있는 파일과 심볼릭 링크 그 자체는 서로 구분하기 힘ㄷ르 정도다.

쉘에서 ls 명령어로 인해 빨간색이나 다른색상으로 깨진 링크를 볼 수 있다.

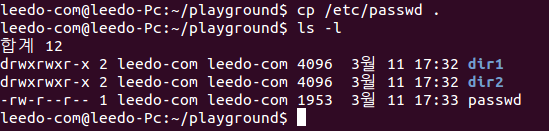
4.7 놀이터를 만들어 보자

연습을 위한 공간(놀이터:playground) 를 디렉토리 공간에 만들어보자.

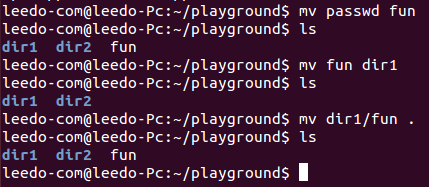
4.7.1 디렉토리 생성



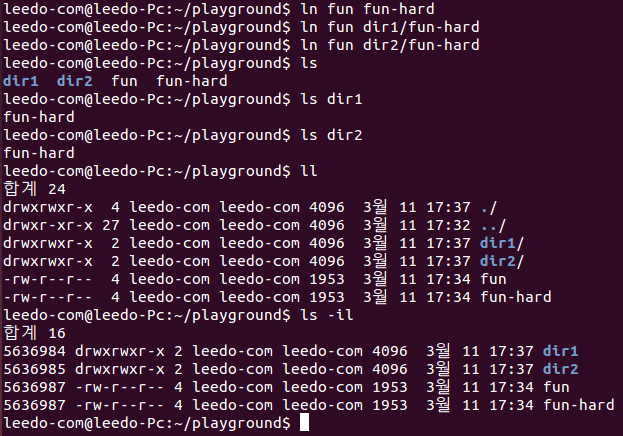
4.7.2 파일복사



4.7.3 파일 이동 및 이름 변경



4.7.4 하드 링크 생성



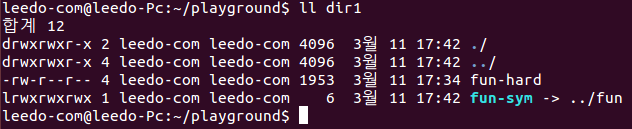
두개의 파일이 동이하다는 사실은

ls -il 으로 아이노두(inode)를 확인해야 가능하다.

4.7.5 심볼릭 링크 생성



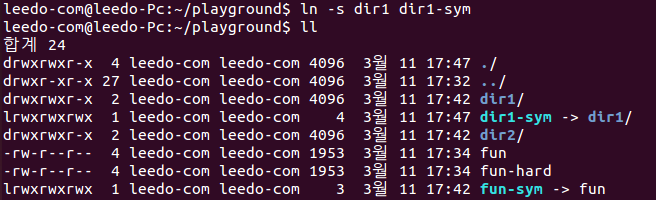
심볼릭 링크를 만드는 것은, 심볼릭 링크가 참조하는 원본파일의 위치 정보를 생성하는 것이다.



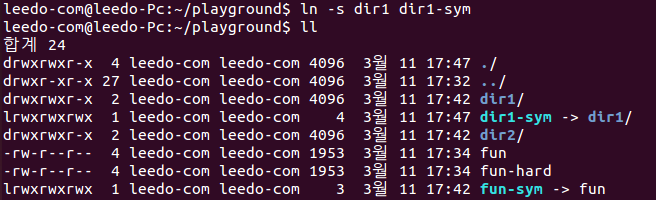
심볼릭 링크 생성시 절대 경로를 사용할 수 있다.

그러나 상대적 경로를 사용하는 것이 바람직하다. 왜나햐면 심볼릭 링크를 포함하고 있는 디렉토리가 링크를 유지하면서 파일명을 변경하거나 파일을 이동할 수 있도록 허용하기 때문이다.

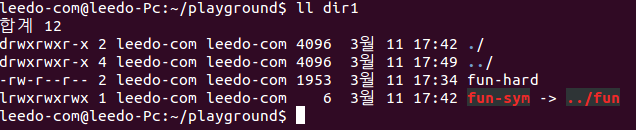
일반적인 파일뿐만 아니라 심볼릭 링크는 디렉토리도 참조할 수 있다.



4.7.6 파일 및 디렉토리 삭제



링크깨짐 발생시 표현



4.8 마무리 노트

마무으리~~~

---------------------------------------------------------------------------------------------------------------------------

5. 명령어와 친해지기

이번 장에 다룰 명령어

type - 명령어의 이름이 어떻게 표시되는지 확인

which - 실행 프로그램의 위치 표시

man - 명령어의 man 페이지 표시

apropos - 적합한 명령어 리스트 표시

info - 명령어 정보 표시

whatis - 명령어에 대한 짧은 설명 표시

alias - 명령어에 별치 붙이기