ex00

man 명령어, 명령어 --help : 해당 명령어의 도움말 제공

mkdir 폴더/ : 디렉토리(=폴더) 생성
mkdir ex{00..10} *// ex00 ~ ex10* 한번에 생성

vim 파일 : 파일 생성 후 vim 편집기 실행
cat > 파일 : 파일 생성, 텍스트를 입력 후 'Ctrl + D'로 저장

cat 파일 : 파일 내용 출력

ex01

touch 파일 : 크기가 0인 파일 생성
touch -t 202406012342 파일 : 파일의 접근시간과 수정시간을 변경

truncate -s 크기 파일 : -s 옵션으로 특정 size의 파일 생성(변경)

ls : 현재 디렉토리의 모든 파일 및 폴더 조회
ls -a : 숨겨진 파일을 포함하여 조회
ls -l : 자세한 정보와 함께 이름순 정렬
-r--r-xr-x *// 퍼미션(권한)*
// 파일 종류: 일반 파일(-), 디렉토리(d), 심볼릭 링크(l)
// 3개 그룹: u(소유자 권한), g(그룹 권한), o(기타 사용자 권한)
// 접근 모드: r(읽기), w(쓰기), x(실행), -(권한없음)

chmod : 퍼미션(권한)을 수정(=change mod)
chmod 453 파일 *// 8진수 모드*
*// 각 그룹 r * 2^2 + w * 2^1 + x * 2^0 (권한이 있으면 1, 없으면 0)*
// r - - (u=4), r - x (g=5), - w x (o=3)
chmod u+r 파일 *// user의 읽기 권한 추가 (기존 권한은 유지)*
chmod g-w 파일 *// group의 쓰기 권한 제거 (기존 권한은 유지)*
chmod o=x 파일 *// others의 실행 권한 지정 (기존 권한은 모두 제거)*
chmod u=rwx, g+rx, o-w 파일
// 유저 rwx권한 지정, 그룹 rx권한 추가, 기타 w권한 제거를 동시에

tar : 파일을 압축하거나 해제
tar -cf 파일명.tar 파일 : 압축파일 생성 (-c = create , -f = file)
tar -xf 파일명.tar : 압축파일 해제 (-x = extract)
tar -xzf 파일명.tar.gz : gzip으로 압축된 파일 해제

ex02

ls -l : 자세한 정보와 함께 이름순 정렬
2 *// 연결된 링크의 수 (하드링크는 2, 심볼릭링크는 1, 디렉토리는 2(자신, 부모))*

하드링크 : 파일의 복사본이 동일한 inode를 가리킴
: 복사본과 원본이 동시에 수정됨
// ln [원본파일명] [하드링크 이름] 로 생성

inode : 파일에 대한 메타데이터를 저장하는 구조체
: 파일을 식별하는 고유한 번호
: 포인터(주소)와 유사
// ls -li 로 확인 가능

소프트 링크 : =심볼릭 링크
: 원본과 복사본이 다른 inode를 가리킴
: 복사본과 원본이 동시에 수정됨
: 디렉토리에 대한 링크도 생성 가능
: 원본 파일의 경로를 참조
: 원본을 삭제하면 연결이 끊김
// ln -s [원본파일명] [소프트링크파일명] 로 생성
// touch -h -t [타임스탬프] [소프트링크파일명] : 로 시간 수정

ex03

ssh-keygen : ssh키 생성 명령어
cd ~./.ssh : .ssh 폴더로 이동
cat id_rsa.pub : id_rsa.pub 내용(=공개키) 출력
// 공개키 복사해서 42intra에 등록
cat id_rsa.pub > id_rsa_pub : 공개키를 id_rsa_pub 파일에 저장

git clone [원격저장소] [로컬저장소] : 깃 레포지터리 연결
git add [폴더] : 깃에서 추적 시작 or 수정사항 스테이징
git add . : 현재디렉토리 및 하위디렉토리의 파일(폴더)을 전부 스테이징
git add —all : 저장소 전체의 모든 파일(폴더)를 스테이징
git add * : 현재디렉토리의 모든 파일(폴더)만 스테이징 (하위디렉토리 제외)

(폴더만 존재하고 안에 파일이 없으면 스테이징 하지않음)

git commit “메시지” : 깃에서 버전 생성(커밋)
git push [원격저장소] : 원격저장소로 제출

ex04

ls -t : 수정 시간 순으로 출력
ls -m : 쉼표로 구분
ls -p : 디렉토리일 경우 /를 추가

ex05

```
#!/bin/bash       // bash 로 쉘스크립트를 실행하겠다는 의미

git log -5 --pretty="%H"
```

// git log : 깃 로그 출력
// -5 : 최근 5개 커밋 조회
// --pretty = ”%H” : 지정한 형식으로 보여줌 (%H: 커밋 해시 (전체))

cat -e : 파일 내용을 출력할 때 출력할 수 없는 문자 표기 + 각 행의 끝에 $를 추가

ex06

```
git ls-files -i --exclude-standard -o
```

.gitignore 파일 : git add에 포함시키지 않을 파일목록

git ls-files : git에 의해 추적되는 모든 파일을 표시
-i (--ignored) : 무시되는 파일에 대한 정보 표시
-o (--others) : -i와 함께 사용, 추적중이지 않은 파일
--exclude-standard: -i와 함께 사용, 표준 깃 제외 항목 표시 (.gitignore 등)

ex07

42intra에서 resources.tar.gz다운
tar xzf resources.tar.gz : 압축해제
.a, sw.diff : mv 또는 cp로 작업폴더로 이동 (._* 파일 : 맥OS에서 생성한 파일일 경우에 생김)

.diff 파일 : 두 파일의 차이점이 기록된 파일
patch a sw.diff : sw.diff 파일을 이용해 a를 업데이트 함
patch a sw.diff -o b : 업데이트한 내용을 -o (--output) 옵션을 이용해서 b파일에 저장

diff a b > sw.diff : a, b 파일의 차이점을 sw.diff파일에 기록
최초 sw.diff 의 내용과 새로생성한 sw.diff의 내용은 같아야함

ex08

```
find . -type f \( -name "*~" -o -name "#*#" \) -print -delete
```

find . : 현재 디렉터리에서 하위 디렉터리 포함해서 파일 찾기

-name: 파일명으로 찾기 (찾으면 true)

"*" 와일드(*)를 사용할때는 문자열로 감싸줘야함
(와일드 카드는 파일 이름의 패턴을 나타내는 특수 문자)

-o : or 연산자
-a : and 연산자
-delete: 검색한 파일 삭제, 단독으로 사용불가(find와 같이 사용)
모든 표현식 사이에는 -a가 생략
단독 표현식에서는 -print가 생략
괄호는 \를 붙여서 사용 \(, \)

```
find . \( -name "*~" -o -name "#*#" \) -print -exec rm {} +
```

-exec : 검색된 파일에서 추가 (외부)명령 실행
{} : -exec옵션에서 사용, find 에서 찾아낸 검색 결과를 명령어 인수로 전달
뒤에 구분자를 붙여서 추가 명령 실행
구분자 + : 모든 표현식의 결과가 연결되어 전체적으로 한 번만 실행
구분자 \; : find를 할때마다 실행

> 참고

```
find . \( -name "*~" -o -name "#*#" \) -print | xargs rm
find . \( -name "*~" -o -name "#*#" \) -print0 | xargs -0 rm
```

명령어 | (파이프라인) : 명령어의 출력을 다른 명령어의 입력으로 전달
-print0 : 검색 결과를 출력하며 검색 항목을 null로 구분
xargs 명령어(rm) : 전달된 입력을 명령어(rm) 인수로 전달
(공백, 탭, 개행문자를 기본 구분자로 사용)
xargs -0 : -0는 xargs가 입력을 NULL 문자로 구분하도록
(파일 이름에 공백, 탭, 개행문자가 포함되어 있는 경우에도 NULL 문자로 안전하게 구분)

ex09

Magic 파일 구조 :

```
41 string 42 42 file
```

[오프셋] [타입] [값] [결과문자열]

// 오프셋 : 파일의 시작위치로부터 얼마나 떨어져있는지를 바이트 단위로 지정 (= 검색할 위치)
: 0부터 시작함
// 42번째(41바이트)

// 타입 : 검색할 타입
// string

// 값 : 검색할 값
// “42”

// 결과문자열 : 검색결과가 일치하면 출력할 문자열 (어떤 타입인지 출력함)
// “42 file”

실제 파일은 바이너리값으로 고유의 매직넘버(시그니처)가 저장됨
.txt 파일은 문자열으로 입력한 내용 자체가 매직넘버로 인식됨

file [파일] : 해당 파일이 어떤 타입인지 출력하는 명령어
file -m [Magic 파일] [파일] : 사용자가 생성한 Magic 파일을 이용해서 타입 출력

—

ETC

리다이렉션
: 쉘에서 명령어의 입력/출력을 파일로 전환하거나, 다른 명령어의 입력/출력으로 연결하는 기능
(파이프라인, 출력 리다이렉션(>/>>), 입력 리다이렉션(<))

명령어 >, >> 파일
: 명령어의 출력을 파일의 내용으로 입력 (>는 덮어쓰기, >>는 이어쓰기)

명령어 < 파일
: 파일의 내용을 명령어의 입력으로 전달

echo "문자열"
: 문자열 출력

cat 파일
: 파일 내용 출력

bash 파일
: 파일의 내용을 명령어로 실행 (= bash 인수로 전달)

명령어 | tee 파일
: 명령어의 출력을 새로만들어진 파일의 내용으로 입력과 동시에 내용 출력
// echo "Hello, World!" | tee output.txt : echo 명령어의 출력을 output.txt에 저장과 동시에 내용 출력
// echo "Hello, World!" > output.txt : echo 명령어의 출력을 output.txt의 내용으로 입력

```
command="ls -l"
eval $command
```
// eval: 변수에 저장된 문자열을 명령어로 실행

입력중이던 내용 지우기
Ctrl + U: 커서 위치에서 줄의 시작까지의 내용을 한 번에 삭제
Ctrl + K: 커서 위치에서 줄의 끝까지의 내용을 삭제
Ctrl + A + Ctrl + K: 줄의 시작으로 이동후 줄의 끝까지 내용 삭제
Ctrl + W: 커서 왼쪽에 있는 단어삭제
Ctrl + C: 실행 중인 명령어를 취소하기