1주차 예비보고서

전공: 수학과,컴퓨터공학과 학년 3학년 학번: 20181256 이름: 김도현

2-1.

홈 디렉토리: /sogang/under/cse20181256

설명

‘~’을 입력하면 홈 디렉토리가 보여진다.( /sogang/under/cse20181256: Is a directory)

2-2.

(데이터 파일)

텍스트이(가) 표시된 사진

자동 생성된 설명

정광민|전라남도 목포시 옥암동 남악29로 301호|013-5649-2092

Tomas|경기도 수원시 영덕대로980 103호|411-3208-7428

Eden|서울시 중구 명동17길 116-12 303호|010-1292-4023

정회준|서울시 용산구 이태원로912 673호|031-7908-4216

김도현|서울시 마포구 만리재로948 914호|015-6707-2937

2-3.

vi data: 입력 시 vi 에디터가 실행되며 ‘data’ 파일이 없을 때는 ‘data’파일이 새로 생성된다. 여기서 ‘data’ 파일을 입력하거나 수정 할 수 있다.

i: 현재는 명령모드이기 때문에 입력이 불가능하다. i 입력 시 커서 앞에서 입력모드로 전환되며 입력을 할 수 있다.

‘Esc’ 키 + ‘:’ : 입력모드에서 콜론모드로 바뀐다.

q!: 콜롬모드에서 q! 입력 시 입력한 데이터가 ‘data’파일에 저장되지 않고 vi 에디터가 종료된다.

w: 콜론모드에서 w 입력 시 입력한 데이터가 ‘data’파일에 저장된다.

wq: 콜론모드에서 wq 입력 시 입력한 데이터가 ‘data’파일에 저장되고 vi 에디터가 종료된다.

2-4.

cp data $HOME/.data

cp data .data(홈 디렉토리에서 실행시)

설명

cp: 이 명령어는 복사를 하기 위한 명령어로 cp data .data 는 ‘data’파일을 ‘.data’파일로 복사하는것이다.

ls -al: 이 명령어를 통하여 현재 디렉토리의 파일의 목록(숨겨진 파일 포함)을 볼 수 있다. ‘.data’파일이 생성되었음을 확인 할 수 있다.

2-5.

chmod 700 $HOME/.data

chmod 700 .data(홈 디렉토리에서 실행시)

설명

chmod: 파일의 권한을 변경하는 명령어이다.

우선 권한 대상자는 소유자(u), 그룹 사용자(g), 다른 사용자(o), 모든 사용자(a)로 나뉜다.

자리는 소유자, 그룹 소유자, 다른 사용자 순으로 되어있다.

모드는 r,w,x(읽기, 쓰기, 실행)로 권한을 2진법으로 표현하여 (111) 모든 권한을 주려면 7이다.

권한을 부여하지 않으려면 (000) 0이다.

즉 700은 소유자만 권한을 부여한다는 뜻이다.

ls -l(ls -al): 이 명령어를 통해서 파일들의 권한을 확인 할 수 있다. (-rwx------)

ls -l .data: 이 명령어를 통해서 ‘.data’ 파일의 권한을 확인 할 수 있다.

2-6.

2-5. 에서 설명했다시피 권한 대상자 u,g,o,a(소유자, 그룹 사용자, 다른 사용자, 모든 사용자)와 모드 r,w,x(읽기, 쓰기, 실행) 권한이 있다.

명령을 수행하기 위해서는 +,-,= 연산자를 통해 수행한다. +는 권한 추가, -는 권한 삭제, =는 권한지정이다.

r 은 읽기 권한으로 파일이나 디렉터리(리스트)를 읽거나 볼 수 있다.

w 는 쓰기 권한으로 파일이나 디렉터리를 쓸 수(수정, 변경, 삭제) 있다.

x 는 실행 권한으로 파일을 실행할 수 있다. 이때 실행 권한은 파일 접근에 필요하기 때문에 읽기 및 쓰기 권한을 하려면 실행 권한부터 있어야 한다.

Ex)

chmod u=r .data: 소유자가 ‘.data’파일의 읽기 권한을 가진다.

chmod u+wx .data: 소유자가 ‘.data’파일의 쓰기, 실행 권한도 가진다.

chmod g=rwx .data: 그룹 사용자가 ‘.data’파일의 읽기, 쓰기, 실행 권한을 가진다.

chmod g-rwx .data: 그룹 사용자가 ‘.data’파일의 읽기, 쓰기, 실행 권한을 잃는다.

3-1.

Regular Expression(정규 표현식, regexp, regex)은 데이터를 검색하거나 패턴들을 매칭하는 문자열에 사용되는 문자이다. 유닉스, 리눅스, 파이썬 등에 사용된다. 기존의 프로그래밍 문법과 느낌이 다르지만 이를 이용할 시 기존보다 문자열을 편하게 다룰 수 있다. PCRE(펄 호환 정규 표현식)라는 표준의 POSIX에서 펄 기반으로 문법을 더 좋고 유연하게 만들어진 정규 표현식도 있다. Regular Expression은 크게 Basic, Interval, Extended 로 나뉜다.

Regular Expression에 이용되는 기호들은 Metacharacters(메타 문자)라고 한다.

Basic Regular Expressions

\a: 일반적인 문자가 아닌 ‘a(특수문자)’를 가진 문자와 매칭한다.

a\*: ‘a’ 를 가지고 있는 문자와 매칭한다.

^a: ‘a’로 시작하는 문자와 매칭한다.

a$: ‘a’로 끝나는 문자와 매칭한다.

. : 어떤 문자든 점의 개수(‘..’ 이면 2개가 대체됨)만큼 대체 가능하다.

a?: ‘a’가 하나만 있는 문자와 매칭한다.

a+: ‘a’가 한개나 그 이상을 가진 문자와 매칭한다.

(): 그룹지어서 Regular Expression을 표현할 수 있다.

Interval Regular Expression

{a}: 앞에 있는 문자와 a번 매칭한다.

{a,b}: 앞에 있는 문자와 a번 매칭하고 b번보다는 작게 매칭한다.

{a, }: {a}와 비슷하지만 a번 이상으로 매칭이 가능하다.

Extended Regular Expression

a₩+b: b앞 a가 있는 문자열과 매칭한다. 이때 a는 1개 이상이 있을 때 매칭한다.

a₩?b: b앞 a가 있는 문자열과 매칭한다. 이때 a는 1개 이하만 있을 때 매칭한다.