1주차 예비보고서

전공: 경영학과 학년: 3학년 학번: 20190963 이름: 한다현

**1.**

유닉스 시스템에 대하여 미리 접해본 후 실험에 임할 수 있도록 한다. 아울러 부록에 나와 있는 명령어에 대하여 익숙해지도록 사용해본다.

**2.**

UNIX 시스템에 접속해본 뒤 자신의 홈 디렉토리를 확인해본다.

홈 디렉토리: /sogang/under/cse20190963

쉘 프로그래밍 실험에서 사용할 데이터 파일인 전화번호부를 만들어본다. 단 데이터 파일의 형식은 실험에서 나온 예제에 따르도록 한다. 5명 이상이 들어가 있는 데이터를 만들되 vi 에디터를 이용하여 작성한다. 단 파일명은 data로 한다.

data:

홍길동|서울시 마포구 신수동 서강대학교 AS관 301호|02-705-2665

박문수|서울시 서대문구 신촌동 연세대학교 연구관 102호|02-708-4678

Andrew|경기도 의정부시 호원동 23-12번지|031-827-7842

한승현|경기도 고양시 일산동구 장항2동 호수마을 407호 1602호|031-904-5643

신봉선|경기도 고양시 일산동구 마두동 206동|031-813-3803

Matthew|서울시 마포구 신수동 서강대학교 MA관 202호|02-777-2000

위의 예제를 편집하는데 사용한 vi 명령어들을 나열하고, 해당 명령 수행하는 결과를 적어보도록 한다.

cd week1: home 디렉토리에서 mydata 파일이 있는 week1 디렉토리로 이동한다.

vi mydata: 실험에서 사용한 mydata 파일을 vi 에디터로 실행한다.

5 + yy: mydata에 5줄의 데이터가 있으므로 첫번째 줄부터 마지막 줄까지 복사한다.

Esc + : + wq: esc와 콜론을 입력하여 mydata 파일의 vi 에디터를 콜론모드로 전환하고 저장 후 종료한다.

vi data: data라는 파일을 생성하고 vi 에디터로 실행한다.

P: mydata 파일에서 복사한 5줄의 데이터를 data 파일에 붙여 넣는다.

i: 커서 바로 앞부터 입력모드로 전환한다. 이후 6번째 줄에서 새로운 데이터를 작성한다.

esc+ : + wq: esc와 콜론을 입력하여 data 파일의 vi 에디터를 콜론모드로 전환하고 저장 후 종료한다.

위에서 작성한 데이터 파일을 $home/.data 파일로 복사한다. 복사하기 위하여 사용한 명령들을 적어보도록 한다.

$HOME: 홈 디렉토리 경로를 확인한다.

cp data /sogang/under/cse20190963/.data: week1 디렉토리에 있는 data 파일을 홈 디렉토리에 .data 파일로 복사한다.

$home/.data 파일을 그룹 및 다른 사용자가 아무 권한도 갖지 않도록 권한 변경을 해본다. 사용한 명령을 적어보도록 한다.

chmod 700 .data

디렉토리에 대한 읽기, 쓰기, 실행 권한을 설정해보고 각각이 갖는 의미를 살펴본다.

디렉토리에 대한 읽기, 쓰기, 실행 권한도 파일에 대한 권한과 동일하게 chmod 명령어를 사용한다. ls -al 명령어를 사용하면 drwx------와 같이 해당 디렉토리나 파일에 대한 권한을 확인할 수 있다. d는 디렉토리를 의미하고 이후 세 글자씩 각각 소요자, 그룹, 다른 사용자가 갖는 권한이다. 이 경우 소유자는 읽기, 쓰기, 실행이 모두 가능하고, 그룹과 다른 사용자는 아무런 권한이 없는 상태이다. r은 4, w는 2, x는 1의 값을 가지며 chmod 명령어를 사용할 때 위의 값들이 합친 숫자를 입력한다. 즉, drwx------의 경우 chmod 700 file(directory)name을 입력한다.

**3.**

정규식은 유닉스 시스템에서 편집 작업을 할 때 유용하게 사용되며 특정 텍스트 문자열을 지정하는 표현식으로 매우 강력한 문자 일치 기능을 갖는다. 또한 정규식을 사용하면 복잡한 검색이나 문자열에 대치 작업도 수월하게 처리가 가능하다. 먼저 임의의 1개 문자를 지정하는 방법은 ‘.’이다. 예를 들어 ‘a.c’라는 문자열이 있다면 이는 a로 시작하고 c로 끝나는 3개의 문자가 있는 문자열이라는 것을 의미하며 abc, adc, aSc 등이 해당된다. 다음으로 []는 문자의 집합을 의미한다. 예를 들어 [abc]는 a, b, c 중 1개 문자를 의미하고 [aA]는 대문자 A나 소문자 a를 의미한다. 이를 확장하면 문자의 범위를 지정할 수 있는데, [0-9]는 0부터 9까지, [A-Za-z]는 1개의 알파벳 대문자나 소문자를 지정하는 것이다. 반대로 [^]는 ^뒤에 있는 문자의 범위에 대응하지 않는 문자열을 검색한다. [^a-z]는 소문자 알파벳이 아닌 문자를 검색하는 것이다. ‘^’기호는 라인의 선두를 의미하여 ^This 형식으로 사용하면 라인의 선두에 등장하는 This 문자열만 찾고 라인 중간에 등장하는 this는 찾지 않는다. 반대로 ‘$’는 라인의 끝을 의미하여 this$와 같은 형태로 사용하면 라인에 끝에 등장하는 this만 찾는다. 또한 ‘\*’는 \* 앞에 지정된 문자와 동일한 문자가 0번 이상 반복되는 문자열을 검색한다. 즉 ‘abc\*’ 형태로 작성하면 ab, abc, abccc 등의 문자열을 찾는다. ‘\<’와 ‘\>’는 각각 단어의 시작과 끝을 의미하며 전자는 특정 문자열로 시작하는 단어가 있는 행을, 후자는 특정 문자열로 끝나는 단어가 있는 행을 찾는다. egrep에서는 더 많은 정규표현식을 지원하는데, ‘+’는 + 앞의 문자가 1번 이상 나타나는 문자열을 검색하고, ‘?’는 ? 앞의 문자가 0번 또는 1번 등장하는 문자열을 검색한다. 또한 ‘|’는 or을 의미하여 ‘apple|banana’의 형태로 작성하면 apple이나 banana가 포함된 행을 검색한다. 마지막으로 ()는 정규표현식을 묶어주는 역할을 하는데, +, ? 등과 같은 연산자는 앞에 등장하는 하나의 문자에 대한 연산자이지만 ()를 사용하여 (hi)+와 같은 형태로 작성하면 hi가 1번 이상 나타나는 행을 검색할 수 있다.