1주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1. 목적**

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

**2. 예비 학습**

UNIX 시스템에 접속해본 뒤 자신의 홈 디렉토리를 확인해본다.

홈 디렉토리 : /sogang/under/cse20212022

쉘 프로그래밍 실험에서 사용할 데이터 파일인 전화번호부를 만들어본다. 단 데이터 파일의 형식

은 실험에서 나온 예제를 따르도록 한다. 5명 이상이 들어가 있는 데이터를 만들되 vi에디터를

이용하여 작성한다. 단 파일명은 data로 한다.

(데이터 파일)

이예준 | 서울시 마포구 신수동 서강대학교 R관 914호 | 010-1234-5678

고길동 | 서울시 도봉구 쌍문동 2-2 | 010-2222-2222

Sherlock | 런던 베이커가 221번지 | +44-20-7224-3688

홍길동 | 서울시 마포구 신수동 서강대학교 K관 202호 | 010-8765-4321

고윤정 | 서울시 영등포구 문래동 | 010-1029-3847

안유진 | 대전시 서구 문산동 | 010-5647-3829

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

적어보도록 한다.

vi data 입력 : data 파일을 만든다.

i 입력 : 명령모드에서 편집모드로 바꾼다.

위의 데이터 파일을 입력한다.

esc 입력 : 편집모드에서 명령모드로 바꾼다.

:wq 입력 : 편집된 파일을 저장 후 vi를 종료한다.

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

cp data $HOME/.data

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

사용한 명령을 적어보도록 한다.

chmod go-rwx data : 그룹(g)과 다른 사용자(o)의 모든 권한(rwx) 삭제(-)

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

chmod는 파일의 그룹, 사용자, 다른 사용자에 대한 읽기, 쓰기, 실행하는 권한을

변경할 수 있는 명령어이다.

chmod 명령어 - chmod <권한 대상> <제거, 삭제 및 지정> <권한 종류> 디렉토리 및 파일

<권한 대상>

1. **u**(User) : 파일 소유자 2. **g**(Group) : 파일이 속한 그룹

3. **o**(Others) : 그 외 사용자 4. **a**(All) : 모든 사용자

<제거, 삭제 및 지정>

1. **-** : 권한 제거 2. **+** : 권한 추가. 3. **=** : 권한 지정

<권한 종류>

1. **r**(read)<4> : 읽기. 2. **w**(write)<2> : 쓰기. 3. **x**(execute)<1> : 실행

예를 들어 **chmod ug +rw [FILE]** 은 파일 소유자, 그룹에게 읽기, 쓰기 권한을

추가하는 명령어이다.

**3. 보충 학습**

Regular Expression에 대하여 정리해보도록 한다.

-Regular Expression 개념

Regular Expression은 일정한 규칙을 가진 문자열의 집합을 표현하는데 사용하는

형식 언어이며 줄여서 Regex, Regexp 또는 정규 표현식이라고도 부른다.

프로그래밍 언어에서 문자열의 검색 및 치환을 위한 용도로 사용하고 있다.

정규 표현식은 크게 POSIX와 PCRE 방식으로 나눠지며, POSIX에서 다시

Basic, Extended로 나눠진다.

- Regular Expression 문법

1. 패턴을 매칭하는 경우

2. 수량을 한정해서 매칭하는 경우

3. 패턴 일부를 그룹으로 묶거나 특정 패턴의 앞뒤로 다른 패턴이 오는 조건을 더하는 경우

- Regular Expression 메타문자

정규 표현식에서 사용하는 기호로 특정한 문자 또는 문자 계열을

대신하여 표현하는 문자이다. 메타 문자를 이용하면 같은 규칙성을 가진 여러 문자열을

하나의 패턴으로 압축할 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| 문자 | 뜻 | 예시 | 맞는 문자열 |
| **^** | 문자의 처음을 나타냄 | ^some | some,something,somebody… |
| **$** | 문자의 끝을 나타냄 | ing$ | sing, listing, something…. |
| **.** | 임의의 한 문자를 나타냄 | a.b | aab, acb, akb… |
| **\*** | 바로 앞에 있는 문자가 없거나  하나 이상이 있음 | rs\*t | rt, rst, rssst…. |
| **+** | 바로 앞에 있는 문자가  하나 이상이 있음 | rs+t | rst, rsst, rssssssst… |
| **?** | 바로 앞에 있는 문자가  없거나 하나 있음 | rs?t | rt, rst |
| **[]** | 한 문자를 가리키며,  괄호 안의 내용은 문자의 범위  **\*특정 문자를 제외하려면 ‘^’사용** | [asdr]re | are, sre, dre, rre |
| [^asdr]re | qre, wre, kre…. |
| **{}** | 앞에 있는 문자의 개수를 나타냄  ‘,’는 문자 개수 범위를 나타냄 | ws{2}p | wssp |
| w{2,}p | wwp, wwwp, wwwwp…. |
| **|** | ‘or’, ‘또는’이라는 의미를 가짐 | pine(apple|cat) | pineapple, pinecat |
| **\** | 메타문자→일반문자로 변환 | [^\.]+\.(set|get) | a.set, k.get |