RedCastle™ for CERT Training 을 이용한

해킹 및 대응 실습

이 교재는 ㈜레드게이트의 해킹 및 대응 훈련시스템인 RedCastle for CERT Training 에 대한 실습교육 정보를 제공합니다

사용자 참고

본 교재는 기술적인 부정확성이나 타이핑 에러를 포함할 수 있다. 변화가 발생시 정기적으로 변경될 수 있다. 이 변화는 개정판에 수록된다. ㈜레드게이트는 본 교재에 기술된 프로그램의 변경 또는 개선을 할 수 있습니다.

목 차

1 UNIX 개요

실험 영역	unix 입문
실험 제목	UNIX 시스템 개요 기본 명령어 학습
실험 요약	UNIX 시스템을 이해하고 기본 명령어를 수행한다.

1.1 UNIX 시스템 개요

1.1.1 UNIX 기원 및 배경

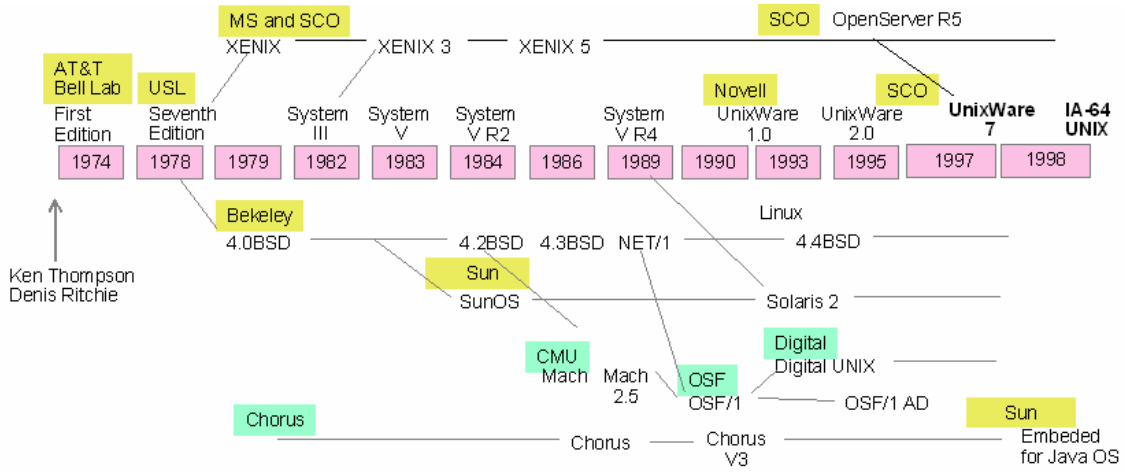
유닉스 시스템은 60 년대 후반, MIT(Massachusetts Institute of Technology), AT&T 社의 Bell 연구소와 당시 컴퓨터 제작사인 GE(General Electronic)社가 “멀틱스(Multics)"라는 실험적인 운영체제에 관한 연구로부터 그 기원을 찾을 수 있다. 1969 년 Bell 연구소의 Ken Thompson 은 ”space travel"이라는 게임을 만들어 이 게임을 DEC 의 PDP-7 에 이식시켰다. 이러한 과정에서 Thompson 과 Ritchie, Rudd Canaday 는 PDP-7 에 대한 새로운 운영체제를 개발하게 되었고 이것이 유닉스의 기원이 되었다.

멀틱스 시스템은 1969 년에 그 초판이 완성되었으나 목적한 수준의 서비스를 제공하지 못하여 Bell 연구소는 멀틱스에 대한 프로젝트에서 손을 떼게되었다. 그러나 멀틱스 프로젝트에 참여했던 몇몇 연구원들이 프로그래밍 환경을 개선하여 최초의 유닉스 버전이 등장하게 되었다.

1973 년 유닉스 운영체제가 C 언어로 작성되면서 그 성능이 개선됨에 따라 점차 유닉스 시스템의 사용이 확산되었고 유닉스 사용자 그룹도 생기게 되었다. 그 후 유닉스는 학술이나 연구 목적으로 기업의 연구 기관이나 대학 등에서 발전을 거듭하여 오늘에 이르게 되었다.

1982 년경 Bell 연구소는 그 동안 개발되어진 유닉스의 변형들을 모아 단일 시스템으로 만들어 유닉스 시스템 III 가 탄생되었고 이후 여러 가지 기능을 추가하여 유닉스 시스템 V 가 개발되었다. 또한 1983 년 캘리포니아 버클리 대학에서 유닉스 시스템의 변형을 발표하였는데 이것이 BSD(Berkeley Software Distribution) 4.3 이다.

아래 그림은 유닉스 계열 운영체제 발전사를 요약하여 보여준다.



<유닉스 발전사>

유닉스는 20 년간 발전을 거듭하면서 많은 계열로 나뉘어졌지만, 현재 유닉스 큰 흐름을 이루는 것은 AT&T 의 "시스템 V" 계열과 버클리 대학의 "BSD" 계열이며 상용 UNIX 에는 Sun Solaris, HP-UX, IRIX, AIX, DIGITAL UNIX, SCO UNIX 등 매우 다양한 UNIX 종류가 있다. 시스템 V 계열 유닉스와 BSD 계열 유닉스 외에도 1980 년 마이크로소프트社에 의해 개인용 컴퓨터에서도 사용 가능하도록 개발된 제닉스(Xenix) 시스템 등이 있다.

리눅스(Linux) 운영체제의 경우 핀란드 헬싱키 대학(Helsinki Univ.)의 학생인 Linus Torvalds 에 의해 개발되으며 1991 년 10 월 버전 0.02 가 발표되어 보급되기 시작했다. 리눅스는 미닉스(Minix)를 기반으로 인텔의 80386 프로세서의 구조를 최대한 활용하고 최적화를 기본 철학으로 만들어졌다. 리눅스는 유닉스를 기본 모델로 하였으며 두 가지 종류로 나뉘어진 유닉스 계열 중에서 SVR4(Unix System V/Release 4) 계열에 치중하여 만들어졌으나 BSD(Berkeley Software Distribution) 특성도 동시에 수용하는 혼합체였다. 또한 IEEE(Institute of Electrical and Electronics Engineers)의 운영체제 인터페이스 표준인 POSIX(Portable Operating System Interface for Computer Environments)를 만족하도록 개발되었다. 이것이 대부분의 유닉스용 프로그램을 리눅스로 이식하기 쉽게 만들었으며 리눅스용 응용 프로그램을 대량 양산하는 역할을 하였다.

1.1.2 UNIX 기본 철학 및 특성

기본적으로 유닉스 시스템은 프로그래밍의 철학이라 할 수 있다. "작은 것이 아름답다(Small is beautiful)."라는 용어를 모토로 유닉스 시스템은 각각의 작은 모듈들이 완전하게 운영될 수 있도록 개발되었다.

유닉스는 지난 20 여 년간 거의 대부분의 컴퓨터를 위한 운영체제로 발전되었다. 기본적으로 유닉스는 광범위한 종류의 기계에 적합하다는 것과 여러 가지의 상이한 환경에 대한 적응력, 사용의 간편함 및 여러 가지의 작업을 동시에 수행할 수 있는 멀티태스킹(Multi-tasking) 기능, 동시에 여러 사용자가 작업할 수 있는 다중 사용자(Multi-user) 환경, 통신망 연결을 위한 환경을 갖추고 있다. 이러한 유닉스의 특성에 대해 알아본다.

첫째, 기종간의 호환성이다. 유닉스 시스템은 대형과 중형컴퓨터에는 이미 널리 사용되어왔고 최근에는 개인용 컴퓨터 수준의 운영체제로도 개발되고 있다.

둘째, 풍부한 기능성을 들 수 있다. 유닉스 시스템의 명령어는 구문이 간단하여 다른 운영체제와는 달리 신속하고, 간단하게 많은 작업을 처리할 수 있다. 다른 시스템에서는 추가의 비용을 지불해야 하는 많은 기능과 명령어가 유닉스 시스템은 기본적으로 갖추고 있다. 그리고 유닉스 시스템의 단점 중 하나였던 GUI(Graphic User Interface)도 최근 많이 개선되었다.

셋째, 풍부한 통신 기능의 지원이다. 다양한 통신 도구나 하위 레벨 장치 드라이버(low-level device driver)들도 추가로 장착하기 쉽게 되어 있으며 최근 윈도 95 나 윈도 NT 등에서 지원하기 시작하는 데이터와 자원공유 기능은 유닉스 시스템에서는 이미 수년 전부터 실용화해 왔었고 높은 기능성과 안정성을 인정받고 있다. 이는 현재 인터넷을 연결하는 시스템의 운영체제로 대부분이 유닉스가 사용되고 있는 것을 보아도 알 수 있다.

1.1.3 UNIX 시스템의 개방성과 그에 따른 문제점

유닉스 시스템은 기본적으로 자원의 공유, 시스템의 공유 등 보안에 대한 고려가 부족하게 설계되어 현재 전자상거래 활성화 등에 따라 정보 보호에 대한 문제가 심각한 고려 대상으로 부각한 시점에서 몇 가지 문제점을 노출시키고 있다. 이러한 유닉스 시스템의 문제점으로 인해 발생 가능한 위협요소를 분류하면 다음과 같다.

가. 차단(Interruption)

시스템의 자원이 소실되거나 사용할 수 없게 되는 경우로서 주로 정보나 프로그램 등을 삭제시키는 경우에 해당된다. 따라서 불법적인 접근 권한 획득에 의한 정보 삭제 및 정보통신망의 하드웨어적 혹은 소프트웨어적 단절, 시스템의 불법적인 파괴에 의한 가동 정지 등이 있을 수 있다.

나. 가로채기(Interception)

불법적인 사용자가 시스템 자원에 대한 접근을 획득하는 경우로서 불법 복사 등이 여기에 해당된다. 주로 통신망을 통해 침입하여 파일을 불법 복제해 가는 경우가 대부분이다.

다. 수정(Modification)

불법적인 사용자가 시스템 자원을 접근할 뿐 만 아니라 변경하는 경우로서 불법적인 읽기와 쓰기가 이루어진다. 운영체제내의 파일 및 프로그램에 대한 불법적인 권한 획득과 함께 이를 이용하여 그 자원의 내용을 변경하는 것을 의미한다.

라. 위조삽입(Fabrification)

불법적인 사용자가 시스템에 위조 정보를 삽입하는 경우로서 주로 데이터베이스나 통신 정보에 위조 정보를 추가하는 경우이다. 즉 단순한 자원의 수정이 아니라 불법적인 정보를 삽입함으로써 구체적인 감사 기능이 없이는 발견하기 어려운 위협이다.

이러한 위협 요소에 따른 침입 유형을 다음과 같이 분류할 수 있다.

가. 스니핑(packet sniffing)

일반적으로 TCP/IP 프로토콜을 사용하는 네트워크에서 Ethernet 을 거치는 데이터는 가공되지 않은 데이터들로서 텍스트 형태로 보내지게 된다. 이러한 패킷들 중에 300 바이트 정도만 가로챌 수 있다고 해도 보안에 큰 타격을 줄 수 있다. 현재 스니퍼(Sniffer)를 위한 소스는 인터넷상에서 손쉽게 구할 수 있고, 다양한 OS 에서 사용 가능하도록 이식되어 있다.

나. NFS(network file system) 문제

네트워크 파일 시스템은 네트워크에 분산된 디스크 없는 시스템이나 디스크가 모자라는 시스템이 서버의 디스크를 공유하도록 하는 것이다. 이러한 환경을 구축할 때 모든 시스템에게 정보가 공개되고 쓰기 권한까지 허가되는 경우가 많다.

다. 네트워크 스캐닝

네트워크 스캐닝은 원격 시스템의 보안 취약성을 검색해주는 프로그램을 이용하여 정보를 수집한 후, 이 정보에 제시된 취약점을 공격하여 비 인가된 접근 권한을 얻는 것이다. 최근에 많이 사용되고 있는 SATAN 은 네트워크에 연결된 호스트들에 관한 여러 정보를 수집하는 시험/보고 도구이다. 특히 SATAN 은 네트워크 서비스 finger, NFS, ftp 등을

실행할 때 호스트와 네트워크에 대한 정보를 수집한다. 그런 다음 요약 형태로 이 정보를 보고하거나 보안 문제를 검사한다.

1.2 유닉스 기본 명령어

UNIX 시스템을 능숙하게 다루고 ,친해지기 위해서는 우선적으로 명령어에 익숙해져야한다. 본 실습에서는 Solaris 서버가 사용되므로 본 장에서는 Solaris 의 기본 명령어를 중심으로 설명하고 있다. 그러나 이러한 명령어는 솔라리스 뿐만 아니라 거의 모든 유닉스 시스템에서 사용되어지는 것으로 기본적으로 꼭 알아야할 것 들이다.

1.2.1 계정 명령어

유닉스에 접속과 접속종료에 관계되는 명령어와 사용자 계정에 관한 명령어에 대해서 알아보자.

가. 로그인

시스템에 접속하기 위해서는 사용자마다 고유의 계정이 있어야한다. 일반 사용자들은 시스템 관리자에게 연락하여 자신이 원하는 ID 로 계정을 발급받아야 한다. 처음 유닉스 시스템에 접속하면 다음과 같은 화면이 나온다.

```
SunOS 5.8
login:
password:
```

사용자는 계정 ID 와 암호를 정확하게 입력하여야한다. 시스템에 성공적으로 접속이 이루어지면, 마지막 login, 전자우편메시지나, 시스템관리자로부터의 메시지 등의 정보가 나온다, 다음은 성공적으로 로긴(login)한 예이다.

```
#Last login: Mon Sep 4 11:19:05 from 192.220.236.8
```

나. 패스워드(Passwd) 바꾸기

시스템에 접속하기 위해서는 매번 로긴과 패스워드를 입력하여야 한다. 만일 패스워드를 잊었다면 시스템 관리자에게 연락하여 패스워드를 재발급 받아야 한다. 사용자에게 있어 패스워드는 자신의 시스템에 대한 1 차적인 보안수단이다. 따라서 정기적으로 바꾸어주는 것이 좋다. 패스워드를 바꾸기 위해서 간단히 passwd 명령을 실행시키면 된다.

```
#passwd
passwd: Changing password for solaris
Enter login password:
New password:
```

다. 로그아웃

시스템 사용이 끝나면 시스템 사용을 끝내겠다는 신호를 시스템에게 보내야하는데 이것을 로그 아웃 또는 로그 오프라고 한다. 쉘에서 로그 아웃하기 위해서는 exit 라는 명령을 입력하면된다. exit 나 키가 입력되면 현재 실행 중인 쉘을 종료시키고 터미날은 사용 대기 상태가 된다.

```
$ exit
Welcome to the AT&T 386 UNIX System
System name: chaerry
login:
```

원격에서 서버에 접속하였을 경우 exit 를 입력하면 접속이 끊어진다.

1.2.2 디렉토리 및 파일처리 명령어

가. 디렉토리 경로명 보기 및 경로 바꾸기

명령어	**pwd 와 cd**
명령어 설명	pwd 명령은 현재 작업 하고 있는 디렉토리의 경로명을 출력한다.
사용법	pwd cd [directory_name]

$ pwd 현재 디렉토리값을 출력한다

$ cd /home/guest /home/guest 디렉토리로 작업영역을 옮긴다.

$ cd 사용자의 홈디렉토리로 이동한다.

$ cd ~ 사용자의 홈디렉토리로 이동한다.(ksh)

$ cd - 바로전에 작업하던 디렉토리로 이동한다. (ksh)

$ cd / root 디렉토리로 이동한다

$ cd .. 현재디렉토리의 바로위 상의 디렉토리로 이동한다.

나. 현재 디렉토리 내용 출력

명령어	**ls**
명령어 설명	ls 명령은 현재 디렉토리 내용을 출력한다. 디폴트는 한 라인에 한 개의 파일명 또는 디렉토리 명칭을 표준 출력으로 출력한다.
사용법	ls [-laixFRtnr] [file_names]

ls 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-A	.과 ..을 제외한 모든 파일과 디렉토리의 리스트를 보여준다.
-a	숨겨진 파일과 디렉토리의 리스트를 보여준다.
-l	파일에 대한 권한, 링크의 수, 그룹, 크기, 마지막 변경시간등을 보여준다.
-R	파일리스트와 디렉토리의 서브디렉토리의 파일까지 보여준다.
-t	가장 최근 것 먼저 정렬한다. (마지막 변경순서로)
-r	정렬 순서를 반대로 한다. (이름은 역순으로 시간은 가장오래된것 먼저 정렬)
-m	파일과 파일사이를 ,(콤마)로 나누어 출력한다.
-i	파일앞에 i-node 의 수를 표시하여 출력한다.
-s	파일과 디렉토리의 크기를 킬로바이트 단위로 출력한다.

ls 명령어에서 a, -l 옵션이 가장 많이 사용되며 아래 그림은 ls l 명령어 사용예를 나타내고 있다.

```
$ ls -l
Total 16
drwxr-xr-x 2 kys other 64 Aug 18 00:34 guest_d
-rwxrwxrwx 1 kys other 40 Aug 18 00:16 samp_1
-rw-r--r-- 1 kys other 256 Aug 18 00:43 sample -> /usr/sample2
-rwxr-xr-x 1 kys other 85 Aug 18 00:55 sp.sh
.....
```

위 예제에서 출력결과는 아래와 같은 의미를 가지고 있다.

option	descrition
Total 16	현 디렉토리 내의 파일들에 의해 사용되고 있는 512 바이트 블럭의 수
-rwxr-xr-x	**파일의 타입(첫번째 문자)** • -일반파일

	• d 디렉토리 • c 문자 디바이스 특수 파일 • b 블럭 디바이스 특수 파일 • p 파이프 특수 파일 • l 심볼릭 링크 파일 **허용모드(나머지 9 개 문자)** • 3 개문자씩 user,group,others 사용자에 대한 파일 허용값이다. user 는 이 파일을 생성한 사용자, group 은 사용자가 포함된 그룹 사용자들에대한, other 는 시스템에 접속하는 모든 사용자들에 대한 파일접근 권한을 설정한다. • rwxr-xr-x 파일의 퍼미션. o r : 파일을 읽을 수 있다 o w : 파일에 기록할 수 있다 o x : 파일을 실행할 수 있다 o - : 파일에 대해 아무것도 할 수 없다
1	파일의 링크 수
kys	파일의 오너(UID). 이 파일의 실제 소유주
other	파일의 그룹(GID). 이 파일의 소유주가 포함되어 있는 그룹명
40	파일의 크기(바이트)
Aug 18 00:16	파일의 생성,수정시간
samp_1	파일명

다. 파일 내용 출력

명령어	**cat**
명령어 설명	지정한 파일을 순차적으로 읽어 들이고, 표준 출력 으로 출력한다
사용법	cat [-options] file1 [file2...]

cat 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-s	: 파일 처리에 대한 에러 메세지를 화면에 출력하지 않는다
-v	: 인쇄 불능 문자를 식별할 수 있도록 출력
-t	: tab 을 ^I 로 출력
-e	: 행 끝에 "$" 문자를 출력

라. 패턴일치 내용 출력

명령어	**grep**
명령어 설명	grep은 Global Reaular Expression & Print의 약어로 지정하는 패턴과 일치하는 내용이 있으면 표준 출력으로 출력하는 명령이다.
사용법	grep [-option] pattern [file(s)]

grep 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition

-c	패턴과 일치하는 라인의 수만 출력한다.
-i	패턴을 검색하는 동안 대문자 소문자를 구분하지 않는다
-l	패턴과 일치하는 라인을 가진 파일의 이름만 화면에 출력한다
-n	패턴과 일치하는 라인의 내용을 라인번호를 붙여서 화면에 출력한다
-v	패턴과 일치하지 않는 라인만 화면에 출력한다.

마. 디렉토리 생성

명령어	**mkdir**
명령어 설명	디렉토리를 생성 하기 위해 생성 하고자 하는 디렉토리로 작업 영역을 변경 한다. 디렉토리 생성시 디폴트 디렉토리 퍼미션은 umask값에 의해 결정되고, 이 값은 옵션을 사용하여 사용자가 지정할 수 있다
사용법	mkdir [-m mode] [-p] directory_name

mkdir 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-m	mode 사용자가 생성하는 디렉토리의 permission mode 를 지정한다
-p	존재하지 않는 모든 부모 디렉토리를 우선 작성한 뒤 dirname 을 생성한다

바. 디렉토리 삭제

명령어	**rmdir**
명령어 설명	rmdir 명령은 삭제 하고자 하는 디렉토리가 비어 있을 경우에만 실행되므로, 디렉토리 삭제를 실행 하기 전에 먼저 디렉토리 내의 모든 파일과 디렉토리를 먼저 삭제해야 한다.
사용법	rmdir [-p][-s] directory_name

rmdir 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-p	/a/b/c 디렉토리를 삭제 하는 경우 a, b,c 디렉토리가 비어 있다면 rmdir 명령은 c b a 순서로 디렉토리를 전부 삭제한다.
-s	-p 항목 지정시 표준 오류(단말기)에 출력되는 메세지를 삭제한다

사. 파일 및 디렉토리 삭제

명령어	**rm**
명령어 설명	파일 또는 디렉토리를 삭제할때 사용되는 명령이다
사용법	rm [-r] [-i][-f] filename

rm 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition

-f	쓰기 퍼미션이 없는 파일을 삭제할 때 사용자에게 확인 을 요구하지 않는다.
-r	인수 리스트에서 지정한 디렉토리, 또는 그 하위의 모든 파일과 서브 디렉토리를 재귀적으로 삭제한다
-i	파일 삭제를 위해 확인이 대화식으로 행해진다. y 값이 입력되어야만 삭제가 실행된다.

아. 파일 및 디렉토리 복사

명령어	**cp**
명령어 설명	한개 또는 복수개의 파일을 다른 파일 또는 디렉토리에 복사한다
사용법	cp [-i][-r] file1 [file2..] target

cp 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-i	타겟 파일이 이미 존재 하고 있는 경우 overwrite 를 할 것인가를 대화식으로 결정한다. y 를 입력하여야 overwrite 된다.
-r	하나의 디렉토리하의 파일들과 또 하부 디렉토리와 파일들을 순차적으로 복사 한다.

사. 파일 및 디렉토리 이동

명령어	**mv**
명령어 설명	한개또는 복수개의 파일을 다른 파일또는 디렉토리로 이동한다.
사용법	mv [-i] file1 [file2..] target

mv 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-i	타겟 파일이 이미 존재 하고 있는 경우 overwrite 를 할 것인가를 대화식으로 결정한다. y 를 입력하여야 overwrite 된다.

아. 링크

명령어	**ln**
명령어 설명	하드링크는 하나의 파일을 여러개의 디렉토리에서 엑세스 할 수 있도록 한다. 실제로 파일을 복사하지 않고 한개의 파일을 여러개의 디렉토리에 다른 이름으로 존재하게한다. 심볼릭 링크는 서로 다른 파일 시스템에 존재하는 파일을 링크한다. 새로운 디렉토리명과 호환을 유지 하기 위해 이전의 디렉 토리명을 가지고 있는다.
사용법	ln file1 [file2..] target ln -s file target

ln 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-i	타겟 파일이 이미 존재 하고 있는 경우 overwrite 를 할 것인가를 대화식으로 결정한다. y 를 입력하여야 overwrite 된다.

자. 파일시스템 정보

명령어	df
명령어 설명	사용가능한 디스크 블록과 free i-node수 출력 Super-block을 검색하여 탑재된 화일시스템 디렉토리,탑재된 자원의 사용 가능한 block및 i-node수를 출력 한다
사용법	df [-lt][-f][file_system|directory mounted_resource]

df 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-l	local file system 에 대해서만 출력
-t	사용가능한 block 및 i-node 뿐만 아니라 할당된 전체 block 과 i-node 가 출력
-f	Super-block 에서 값을 취하는것이 아니라 free-list 내의 정확한 block 수가 출력

차. 파일 및 디렉토리 검색

명령어	**find**
명령어 설명	화일과 디렉토리 검색 지정한 디렉토리부터 디렉토리 계층을 재귀적으로 논리식과 일치하는 화일을 찾아 내려간다. +n은 n보다큰값, -n은 n보다 작은 값, n은 n값을 의미한다.
사용법	find pathname expression

find 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-i	타겟 파일이 이미 존재 하고 있는 경우 overwrite 를 할 것인가를 대화식으로 결정한다. y 를 입력하여야 overwrite 된다.
-name	file 지정한 화일의 화일명과 일치하는 화일
-type c	file type 이 c 인 경우의 화일 • b : block special file • c : character special file • d : directory • p : fifo • f : general file
-links n	file 에 n 개의 link 가 있는 경우의 화일
-user uname	file 의 user name 이 uname 인 화일
-group	file 의 group name 이 gname 인 화일

gname	
-size n[c]	file 의 길이가 n blocks 인 화일. n 뒤에 c 가 있으면 크기는 문자수
-atime n	file 이 n 날자 이내에 액세스된 경우의 화일
-mtime n	file 이 n 날자 이내에 수정된 경우의 화일
-print	현재의 경로명으로 출력
-newer file	현재의 화일이 지정인 수 화일 보다 최근에 수정된 경우의 화일
-exec command	지정한 command 를 실행 한다.인수{ }은 \(expression\)에서 검색중인 경로이름으로 대체된다.
Expression	• !expression 표현식의 부정 • expression1 expression2.. 표현식의 AND 값 • expression -o expression2.. 표현식의 OR 값

카. 파일 비교

명령어	diff
명령어 설명	화일의 차이비교 지정된 두개의 화일 내용을 일치 시키기 위해 변경이 필요한 행을 출력
사용법	diff [-efbh] file1 file2

diff 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-b	행의뒤에 계속되는 공백은 무시되고 다른공백의 문자열은 동등한 것으로 간주한다
-e	file1 을 file2 로 재작성하는 편집기 ed 를 위한 a,c,d 명령어의 절차를 만든다
-f	e 와 비슷한 절차를 만들지만 순서가 반대이기 때문에 ed 를 이용할수 없다
-h	고속이지만 정밀한 일을 처리하지는 못한다. 길이가 무제한인 화일을 처리할 수 있지만 변경 범위가 좁고 확실히 구분 되는 경우에만 사용할수 있다

타. 파일 속성 변경

명령어	chmod
명령어 설명	화일 과 디렉토리에 대한 조작 허용 모드 변경 지정된 화일 또는 디렉토리의 permission mode를 변경. mode변경은 8진수를 사용한 절대수 모드 또는 수식 형식의 오프레이션 모드로 할 수 있다.
사용법	chmod mode file.. | directory.. chmod aaa operator bbb file(s)

o 절대수모드로 퍼미션을 변경하는경우 사용하는 값:

- 1000 sticky bit 설정
- 0400 사용자에 의한 읽기
- 0200 사용자에 의한 쓰기

- 0100 사용자에 의한 실행
- 0070 그룹에대한 읽기,쓰기,실행
- 0007 일반 사용자에의한 읽기,쓰시,실행

o Operation 모드로 퍼미션을 변경하는 경우 사용하는 값:

- 사용형식 aaa +- bbb

  aaa : u(화일이나 디렉토리의 주인인 사용자), g(그룹), o(일반 사용자)

  a(모든 사용자)

  operator : +(permission 첨가), -(permission 삭제), =(절대수로 지정)

  bbb : 허가유형을 가리키는 1 자이상의 Permission 문자.

  r (읽기가능) , w (쓰기가능) , x (실행가능) , s (사용자(SUID) 또는 그룹설정(SGID) ID 상태로), t (sticky bit 설정), l (액세스하는동안 강제 lock 발생)

chomde 에 대한 사용예는 아래와 같다.

$ chmod g+wx sample

sample 파일의 그룹 사용자들에게 쓰기와 실행 허용을 추가한다.

$ chmod 777 sample

sample 파일에 대해 모든 사용자들이 읽고,쓰고,실행가능하게 한다.

파. 소유자 또는 그룹 변경

명령어	chown, chgrp
명령어 설명	소유자 또는 그룹의변경 화일이나 디렉토리의 소유자와 그룹을 변경할 수 있는 명령이다. 어떤 화일에 대한 소유자명을 새로운 사용자명으로 재설정할 수 있는 명령이다. 이 명령이 실행 되고 나면 그 화일에 대해 이전의 소유자는 이 화일에 대해 이 명령을 다시 실행할 수 없다. 시스템 보안을 위해 수퍼바이져만 실행할 수 있도록 명령을 실행을 허용하는 경우가 많다.
사용법	chown owner file|directory chgrp group file|directory

chown 및 chgrp 사용예는 아래와 같다.

$ chown root sample

sample 파일의 생성자를 root 로 변경한다.

$ chgrp src sample

sample 파일의 그룹명을 src 로 변경한다.

1.2.3 시스템 정보

가. 프로세스 상태

명령어	ps
명령어 설명	현재 실행 중인 프로세스의 상태를 출력하는 명령이다. 아무런 옵션이 설정되지 않은 경우는 현재 사용중인 단말기와 관련된 프로세스 정보만 출력한다
사용법	ps [-options]

ps 명령어에서 사용되는 옵션은 아래와 같다.

option	descrition
-e	현재 실행중인 모든 프로세스에 관한 정보를 출력
-d	프로세스 그룹의 리더를 제외한 모든 프로세스에 관한 정보 를 출력한다.
-a	가장 빈번하게 요구되는 모드 프로세스에 관한 정보를 출력
-f	프로세스 상태를 full list 로 출력한다
-l	프로세스 상태를 long list 로 출력한다
-t tty_no	지정하는 단말에서 실행된 프로세스 정보를 출력한다
-p PID	지정하는 프로세스 번호를 가진 프로세스 정보를 출력
-u UID	지정하는 사용자 번호(UID)를 가진 프로세스 정보를 출력
-g GID	지정하는 그룹 번호(GID)를 가진 프로세스 정보를 출력

아래의 ps -l 명령어 수행시 출력예를 나타내고 있다.

```
F  S  UID  GID  PID  PPID  C  PRI  NI   ADDR     SZ     WCHAN    TTY   TIME   COMD
8  0  310   1   183   170  0   67   20  80002A32  142  EA0242D2  pts/3  00:01   sam
```

option	descrition
F	process 에 관련된 flag • 00 프로세스 종료 • 01 시스템 프로세스. 항상 메인 메모리에 상주 • 02 부모 프로세스가 추적 프로세스 • 04 추적 부모 프로세스의 신호에 의해 정지되어있다 • 08 프로세스가 현재 메인 메모리에 있음

	• 10 프로세스는 메인 메모리에 있고 lock 되어 있다
S	프로세스 상태 • O 현재 실행중 • S 휴식상태, 어떤 작업이 종료되기를 기다리고 있다 • R 실행가능한 상태, 프로세스가 실행 대기 행렬에서 대기 하고 있다 • I Idle 상태 • Z Zombie 상태. 프로세스는 종료되었지만 부모프로세스가 대기하고 있는 상태가 아니다 • T 추적상태. 추적하고 있는 부모 프로세스에 의해 프로세 스가 정지됨 • X SXBRK 상태. 프로세스는 좀더 많은 메모리를 할당 받기 위해 대기 중인 상태
UID	사용자 ID 또는 사용자명
GID	그룹 ID
PID	프로세스 ID
PPID	부모 프로세스 ID
C	스캐줄링을 위한 프로세스 소모량
PRI	프로세스의 우선순위.
NICE	프로세스의 우선 순위값을 조절하는 nice 의값(디폴트 20 이다)
ADDR	프로세스의 메모리 주소
SZ	메인 메모리내에 교체 가능한 프로세스 이미지의 크기 (page 수), pagesize 명령을 사용하면 한 페이지의 크기를 알수 있다. 보통은 1page=4Kbytes 이다.
WCHAN	sleep 또는 SXBRK 상태로 되어 있는 프로세스가 대상으로 있는 이벤트의 주소(이 주소가 공백인 경우 그 프로세스 는 현재 실행중이다
TTY	실행 터미널 번호
TIME	프로세스의 누적 실행 시간
COMD	실행되고 있는 프로그램명

나. 프로세스 종료

명령어	kill
명령어 설명	프로세스종료 또는 프로세스에 시그널 전달 지정된 프로세스에 대해서 신호 15(종료신호)를 보내어 프로세스를 종료 시킨다. 신호를 처리하지 않거나 무시하지 않는 프로세스 종료를 위해 9 라는 시그날 값을 지정하여 강제 종료 시킬수도 있다.
사용법	kill [-signo] PID..

kill 명령어의 사용예는 아래와 같다.

$ kill -9 242

PID 242 프로세스 종료

$ kill -HUP inetd

inetd 데몬 프로세스에게 hangup 신호를 전달하여, 변경된 inetd.conf 파일을 다시 읽도록 지시한다.

1.2.4 네트워크 사용 기본 명령어

가. 원격 접속 명령어

명령어	**telnet**
명령어 설명	네트웍에 연결된 다른 컴퓨터에 단말기로 접속할 수 있게하는 응용 프로그램이다.
사용법	telnet [호스트] [포트]

여기서 호스트는 접속할 호스트를 의미하며 인터넷 주소형식으로 사용된다. 또한 포트는 접속에 이용할 호스트의 서비스 포트를 의미한다. 포트가 지정되지 않으면, telnet 에서 사용되는 초기값이 사용된다.(23 번 포트)

```
$telnet 192.168.1.102
Trying 192.168.1.102...
Connected to 192.168.1.102.
Escape character is '^]'.
Authorized use only.

HogWat. Testing Server..

login: castle
Password:
Last login: Mon Aug 18 15:46:58 from 192.168.1.10
Sun Microsystems Inc.   SunOS 5.8       Generic Patch   October 2001
HogWat:/export/home/castle>
```

여기에서 " ^] " Key 는 local 시스템으로 돌아온 뒤 telnet 프롬프트를 출력한다. 다음은 " telnet> " 에서 사용할수 있는 명령어를 요약하면 다음과 같다.

command	descrition
close	open 되어있는 telnet session 을 close 하고 exit 를 실행
open host-name	지정한 hosts 와 telnet 포트로 연결한다
quit	exit telnet
send arguments	한 개 또는 여러개의 특수 문자를 remote host 에 전송한다. escape, b 가, ip, ao,....

위 명령어에 대한 사용예는 아래와 같다.

```
$telnet
```

```
telnet> open 192.168.1.105
Trying 192.168.1.105...
Connected to 192.168.1.105.
Escape character is '^]'.

SunOS 5.8

login: '^]'
telnet> quit
Connection closed.
$
```

나. 원격 파일전송 명령어

명령어	**ftp**
명령어 설명	remote 호스트와의 파일 전송 또는 수신을 지원한다. ftp 는 FTP(File Transfer Protocol)프로토콜을 이용하는 응용 프로그램이다.
사용법	ftp [-options] [host [port]]

여기서 호스트는 접속할 호스트를 의미하며 인터넷 주소형식으로 사용된다. 또한 포트는 접속에 이용할 호스트의 서비스 포트를 의미한다. 포트가 지정되지 않으면, telnet 에서 사용되는 초기값이 사용된다.(21 번 포트)

```
$ftp 192.168.1.102
Connected to 192.168.1.102.
220 HogWat FTP server (WARNING:Authorized use only) ready.
Name (192.168.1.102:castle): castle
331 Password required for castle.
Password:
230 User castle logged in.
ftp>
```

ftp 명령어를 이용하여 remote 호스트에 접속하는 절차이다. 모든 절차가 정상적으로 종료되면 사용자는 ftp> 프롬프트를 받게 된다. ftp> 프롬프트상에서 사용할 수 있는 명령어는 다음과 같다.

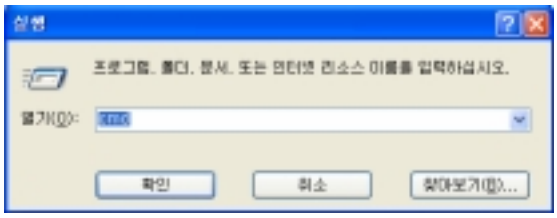
command	descrition
open host 명	지정한 hosts(ftp server)와 연결을 설정한다
close	연결을 종료한다
bye	remote server 의 ftp session 을 종료하고 exit ftp 한다
exit	연결을 종료한다
! [command]	local machine 상에서 shell 명령어를 실행한다
ls	remote machine 상의 디렉토리 리스트를 출력

cd directory	remote machine 상에서 디렉토리 변경
lcd dir.	local machine 상에서 디렉토리 변경
get file	remote file 을 local machine 의 현재 디렉토리로 가져온다
mget files	remote file 복수개를 local machine 의 현재 디렉토리에 대화식으로 가져온다
put file	local file 을 remote machine 에 저장한다
mput files	복수개의 local-files 을 remote machine 상에 저장
hash	파일 전송 진행 정도를 표시하기 위한 명령. 일반적으로 데이터 송수신에 #이 표시된다.
ascii	아스키 문자 파일을 송수신할 때 지정 한다
binary	binary 파일을 송수신할 때 지정한다
?	사용할 수 있는 명령에 대한 정보를 출력한다

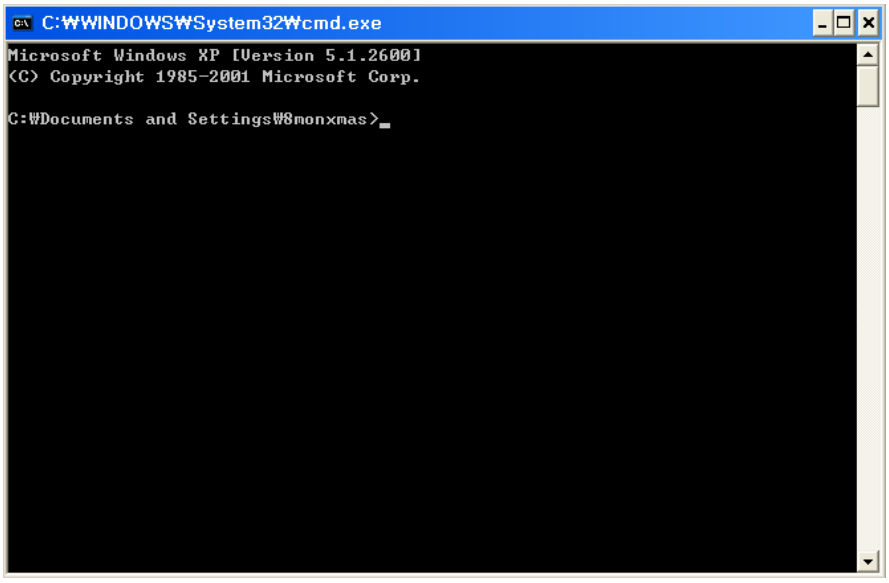
다. PC 에서의 telnet 및 ftp 사용

(1) 명령어 창을 이용

PC 에서는 명령어 창을 이용할 경우 '시작' 메뉴에서 '실행'을 클릭하면 아래 창이 나타난다.



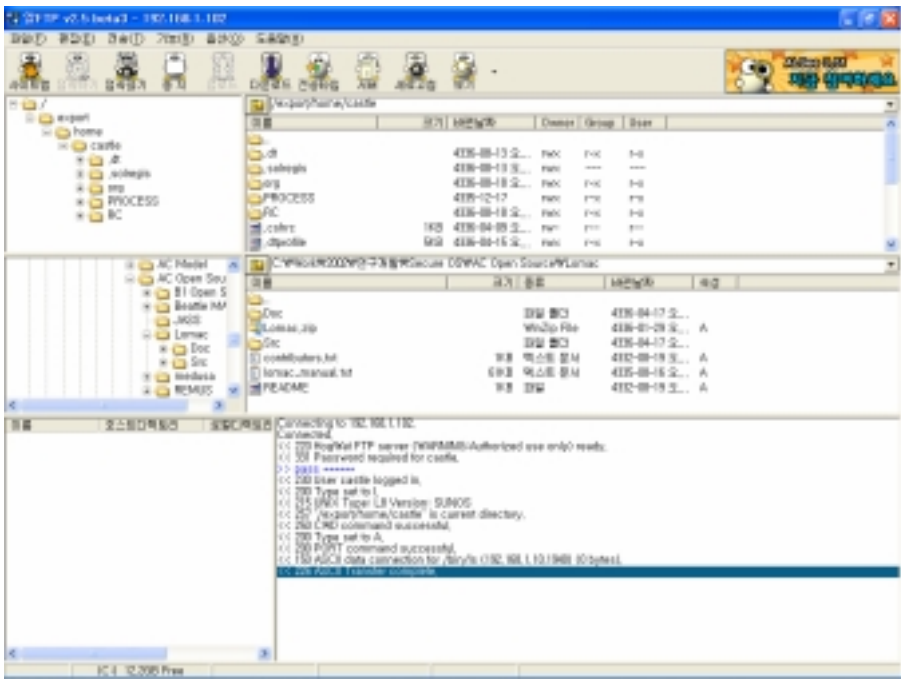
위 실행 창에서 열기' 에 'cmd'를 입력하면 아래와 같이 명령어 창이 나타난다.



실습자는 여기에서 telnet 또는 ftp 명령어를 수행할 수 있다.

(2) telnet 및 ftp 에뮬레이터 사용

Windows 기반의 PC 환경에서는 telnet 및 ftp 에뮬레이터를 이용하여 손쉽게 telnet 이나 ftp 를 사용할 수 있다. 아래는 개인의 경우 무료로 사용할 수 있는 '알ftp'의 사용예를 나타내고 있다.



1.3 텍스트 편집기

1.3.1 텍스트 처리 도구(vi)

vi 편집기는 주로 UNIX 시스템에서 사용할 수 있도록 만들어진 Interactive Text Editor 로서 ex 편집기에 기초를 둔 display 지향 편집기이며 사용자는 화일 편집을 편리하게 할수 있는 기능을 부여 받는다.

Vi 편집기의 특징은 아래와 같다.

- 커서의 이동이 자유롭다
- 화면의 편집 위치가 자유롭다

- 편집 기능이 다양하다
- 화면 조정이 자유로운것을 들 수 있다.

vi 편집 모드에는 명령어 모드와 텍스트 입력 모드가 있다. 먼저 명령어 모드는 vi 에디터에 처음 들어가게 되면 사용자는 명령어 모드에 있다. 이때 입력하는 문자는 vi 에 의해 명령어로 인식된다. 즉 커서를 이동한다던가 화면을 넘긴다던가, 라인을 삭제하거나, 복사하는등의 명령을 실행할 수 있다. 다음으로 텍스트 입력 모드는 i,a,o 와 같은 문자를 입력하는 명령을 주고나면 그때 부터 입력되는 모든 문자는 화면에 display 되고 버퍼에 입력으로 들어가게 된다. 입력 모드에서 빠져 나오기 위해서는 키를 한번 눌러 준다

1.3.2 vi 편집기 사용 방법

$ vi vitest

vitest 라는 화일을 vi 편집기 안으로 불러 들인다.

이화일 이미 존재하는 화일이면 화일 내용이 화면에 display 되고 화면 마지막 라인에 화일명, 라인수,문자수를 출력한다

$ vi -r vitest

편집 도중 시스템 장애로 인하여 저장하지 않고 종료된 편집 화일을 다시 불러 사용할 수 있는 옵션이다

$ vi +10 vitest

vitest 화일의 10 번 라인을 첫번째 컬럼에 커서가 이동되고 편집이 시작된다

가. vi 편집기에서 빠져 나오기

vi 편집기를 호출한 뒤 화일의 내용을 수정하고 편집기에서 빠져 나오는 명령들이다

option	descrition
ZZ	버퍼에 있는 내용을 디스크에 저장하고 vi 에서 빠져 나온다
:wq	ZZ 과 같다
:w	버퍼의 내용을 디스크에 저장한다
:w filename	버퍼의 내용을 지정한 화일이름으로 디스크에 저장한다

:w! filename	버퍼의 내용을 기존에 있는 화일명으로 디스크에 저장한다
:q	vi 에서 빠져나온다
:q!	현재까지 편집한 내용을 무효화하고 빠져나온다

나. 화면 이동

option	descrition
Ctrl-f	한 화면 아래로 이동
Ctrl-b	한 화면 위로 이동
Ctrl-d	반 화면 아래로 이동
Ctrl-u	반 화면 위로 이동
Ctrl-e	한 라인 아래로 이동
Ctrl-y	한 라인 위로 이동

다. 커서 이동

option	descrition
<Enter>	다음 라인의 첫번째 칼럼
h	한 문자 왼쪽으로 이동
j	다음 라인의 같은 컬럼으로 이동
k	위 라인의 같은 컬럼으로 이동
l	한 문자 오른쪽으로 이동
w	다음 단어의 첫자로 이동
e	현재 커서가 위치한 단어의 끝으로 이동
b	바로 이전 단어의 첫자로 이동
^	현재 위치에서 그 라인의 첫번째 컬럼으로 이동
$	현재 위치에서 그 라인의 마지막 컬럼으로 이동
H	현재 화면의 첫번째 라인의 첫 컬럼으로 이동
M	현재 화면의 가운데 라인의 첫 컬럼으로 이동
L	현재 화면의 마지막 라인의 첫 컬럼으로 이동
:n	버퍼내에서 n 번째 라인으로 커서를 이동
nG	버퍼내에서 n 번째 라인으로 커서를 이동

라. 문자열 검색

option	descrition
/pattern	현재 커서가 있는 위치에서 아래 방향으로 pattern 과 일치하는 문자열을 검색한다
?pattern	현재 커서가 있는 위치에서 윗 방향으로 pattern 과 일치하는 문자열을 검색한다
n	문자열 검색을 입력된 검색 명령과 같은방향으로 반복한다
N	문자열 검색을 입력된 검색 명령과 반대방향으로 반복한다

마. 문자 또는 새로운 라인 추가

option	descrition
i	현재 커서가 위치한 앞자리에 새로운 문자나 문장을 삽입한다. (Insert)
a	현재 커서가 위치한 뒷자리에 새로운 문자나 문장을 삽입한다. (Append)

바 새로운 라인 삽입

option	descrition
I	현재 커서가 위치하고 있는 라인의 앞에 새로운내용 삽입
o	현재 라인의 다음 라인을 비워서 내용 삽입
O	현재 라인의 윗 라인을 비워서 내용 삽입

사 문자 및 단어 치환

option	descrition
r	현재 한 문자를 다른 문자로 치환
R	현재 위치의 임의 갯수문자를 다른 문자로 치환
cw	현재 커서 위치에서 그 단어의 끝까지를 바꾼다

아. 라인 치환

option	descrition
cc	한 라인을 임의의 다른 내용으로 변경
C	현재 커서의 위치에서 그 라인의 끝까지를 임의의 다른 내용으로 변경
s	현재 커서의 위치에서 원하는 문자열 만큼 다른 문자열로 치환

바. 삭제

option	descrition
x	커서가 위치한 한 문자를 삭제
X	커서가 위치한 바로 앞 문자를 삭제
dd	커서가 위치한 라인을 삭제
D	현재 커서 위치에서 그 라인의 끝까지 삭제
dw	현재 커서가 위치하고 있는 단어의 뒷부분과 다음 단어 앞의 공백을 없애고 다음 단어를현재 단어의 나머지 부분에 합친다
db	현재 커서 위치에서 왼쪽으로 한 단어 삭제

사. 복사 및 이동

option	descrition
yw	현재 커서가 위치한 단어를 임시 버퍼에 복사
yy	현재 커서가 위치한 라인을 임시 버퍼에 복사
nY	현재 커서가 위치한 라인부터 n 개 라인을 임시 버퍼에 복사
p	현재 커서의 오른쪽 또는 아래 라인에 임시버퍼에 들어 있던 내용을 put
P	현재 커서의 왼쪽 또는 윗 라인에 임시 버퍼에 들어 있던 내용을 put
u	바로 전에 행한 명령의 실행 취소

U	바로 전 한라인에 대한 모든 명령의 실행 취소

아. 라인 결합 및 분리

option	descrition
J	현재 라인과 다음 라인을 한 라인 으로 결합
i	한 라인을 현재 커서 위치를 기준으로 두 개라인으로 분리

1.4 실험

본 실험에서는 UNIX 시스템에 대한 기본 명령어를 이용하고 이를 수행해본다. 실험 전 먼저 Windows 환경의 PC 에서 실험 서버로 telent 을 이용하여 접속하여야 한다. telnet 접속을 위하여 명령어 창을 이용하여도 되며 telnet 에뮬레이터를 이용하여도 된다.

※ 주의사항 : 실험에 사용되는 서버는 운영체제보안 소프트웨어가 설치되어 있으며 주요 시스템 파일에 대한 불법적인 변경으로부터 서버를 보호한다. 일반 UNIX 시스템과 같이 모든 파일을 읽거나 명령어를 수행할 수 있지만 실험에서 허가된 파일 또는 디렉토리만 변경이 가능하다. 본 실험에서는 파일 및 디렉토리의 생성, 편집, 삭제 등은 사용자 홈디렉토리 이하에서만 허용한다.

※ 사용자는 ID 는 user001 부터 user100 까지 지정되어 있으며 사용자 패스워드는 최초 사용자 ID 동일하게 설정되어 있다.

가. 현재 작업하고 있는 디렉토리명을 출력하시오

나. /etc 디렉토리로 이동하시오

다. /etc/passwd 파일의 내용을 출력하시오

라. /etc/password 파일에서 사용자 root 의 정보만 출력하시오

마. 사용자 홈디렉토리로 이동하고 사용자 홈디렉토리에 test 디렉토리를 만드시오.

바. 사용자 홈디렉토리의 local.cshrc 파일을 test 디렉토리로 test.txt 파일로 복사하시오.

사. test 디렉토리의 test.txt 파일명을 test.tmp 로 변경하시오

아. test 디렉토리의 test.tmp 파일을 사용자 홈디렉토리에서 직접 접근할 수 있도록 링크 파일을 만드시오

자. 사용자 홈디렉토리의 local.cshrc 의 속성을 rwxrwxrwx 로 변경하시오

차. 사용자 홈디렉토리에 잇는 test.tmp 파일을 삭제하시오

카. 사용자 홈디렉토리에 있는 test 디렉토리를 삭제하시오

타. /bin/proctest 를 실행하고 프로세스 정보를 출력하시오

파. protest 프로세스를 중지하시오

하. vi 에디터를 사용하여 아래 문장을 test.txt 파일로 만드시오

```
# @(#)inetd.conf 1.23 90/01/03 SMI
#
# Configuration file for inetd(8). See inetd.conf(5).
#
# Internet services syntax:
# <service_name> <socket_type> <proto> <flags> <user> <server_pathname> <args>
# Ftp and telnet are standard Internet services.
# ftp stream tcp nowait root /usr/etc/in.ftpd in.ftpd
telnet stream tcp nowait root /usr/etc/in.telnetd in.telnetd
nntp stream tcp nowait root /etc/nntpd nntpd
```