

Unix & Shell

한양대학교 컴퓨터공학과

***Dept. of Computer Science & Engineering
Hanyang University***

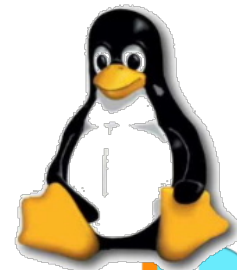
Unix & Linux

- Unix

- 1969년 AT&T의 지부인 Bell연구소에서 Ken Thompson에 의해 개발
- 현재 “Unix”는 POSIX 표준을 충족시키는 운영체제를 설명하는 용어로 지칭

- Linux

- 1989년 핀란드 대학생인 Linus Torvalds가 유닉스를 기반으로 개발한 공개 운영체제
- 워크스테이션이나 개인용 컴퓨터에서 주로 사용되는 유닉스와 유사한 운영체제
- 소스코드를 완전 무료로 공개



Unix & Linux 의 특징

- 멀티태스킹(Multitasking)
 - 시스템이 한번에 하나 이상의 프로그램을 실행 가능
- 멀티유저(Multiuser)
 - 한 시스템에 다수의 사용자가 동시에 접속하여 사용 가능

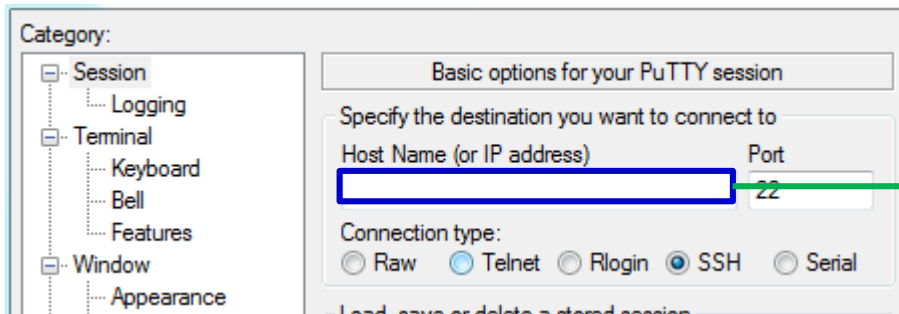


ID and Password

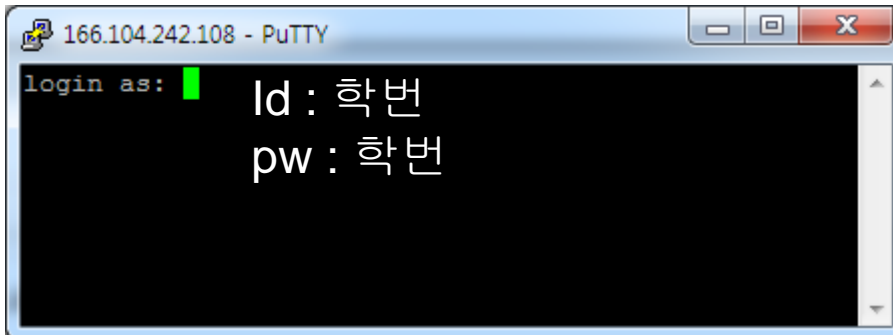
- 시스템 관리자(System Administrator, Super User, Root)
 - 운영체제의 관리와 유지/보수를 수행하는 사람
 - 프롬프트 상 표기 : #
- 사용자 계정(User id, Account)
 - 사용자를 구분할 수 있는 고유한 이름
 - 프롬프트 상 표기 : \$ or % (셸 종류에 따라 다르게 표기됨)
- 사용자 암호>Password)
 - 시스템을 사용할 허가

Telnet & SSH Client

- Putty
 - 무료 텔넷 접속 프로그램
 - <http://kldp.net/projects/putty/download>



접속할 서버의 IP 입력
(166.104.246.108)



Change Password

- passwd 명령어를 사용
 - 사용자명이 "hycse"인 사용자가 패스워드를 변경하는 과정

```
$ passwd hycse
Changing passwd for hycse
Old password:
New password:
Re-enter new password:
$
```

How to logout

- 작업을 마치려면 로그아웃 명령으로 세션을 마쳐야 한다
- 로그아웃
 - 유닉스에게 현재의 사용자명으로 로그인한 사용자가 작업을 마친다고 알리는 것
- 로그아웃 방법
 - 쉘 프롬프트가 나타날 때까지 기다렸다가 Ctrl-D키를 누름
 - logout 명령어를 사용
 - exit 명령어를 사용

사용자 계정 생성 및 삭제 (관리자모드)

- 계정 생성

- adduser or useradd 명령어 사용

```
# adduser username
```

- 비밀번호 변경

- passwd 명령어 사용

```
# passwd username
```

- 계정 삭제

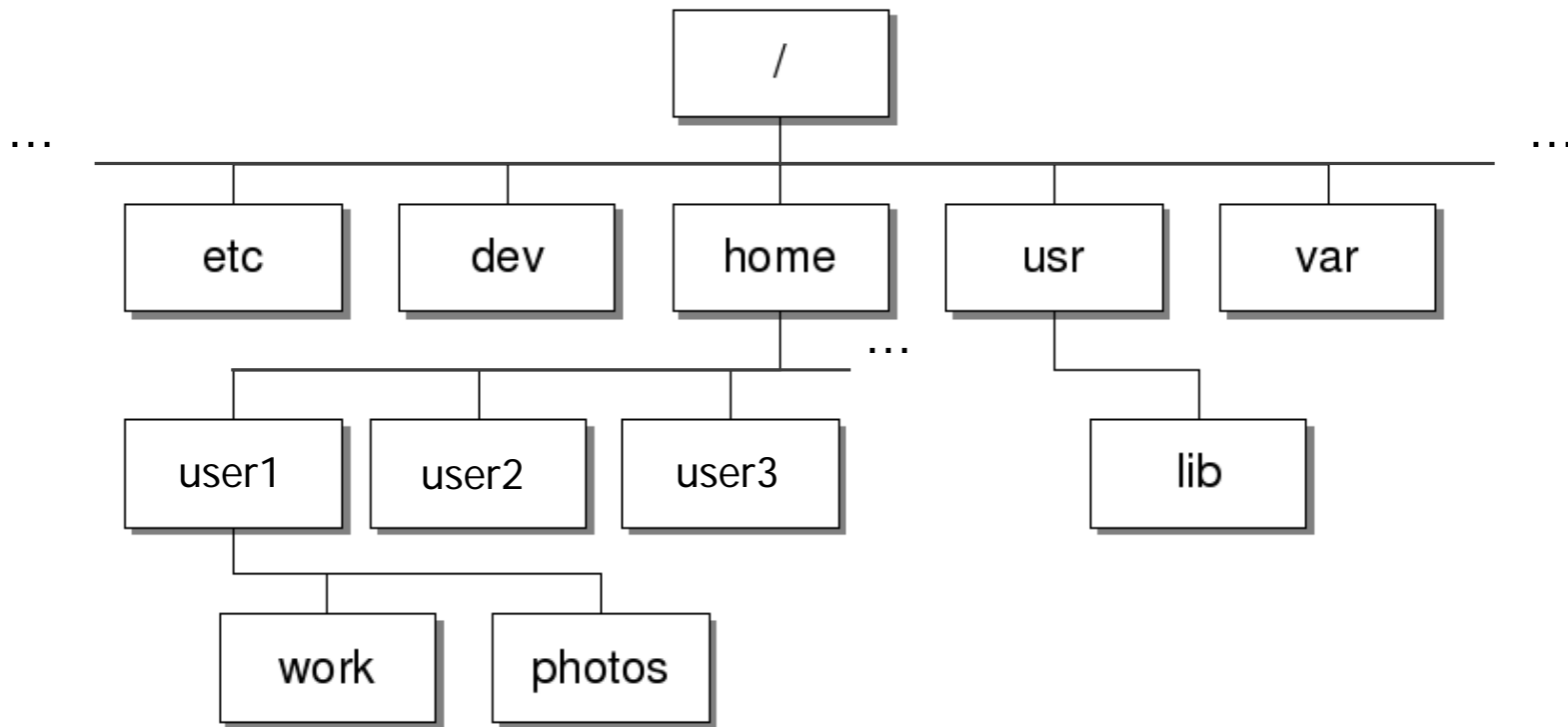
- deluser(or userdel) 명령어 사용

```
# deluser username
```


리눅스 파일시스템

디렉토리명	의 미
/	최상위 디렉토리
/bin	기본적인 실행 가능한 명령 파일을 갖는다
/dev	특수 파일인 장치 파일을 갖는다
/etc	시스템을 관리하는데 필요한 파일을 갖는다
/tmp	시스템이 임시 저장용으로 사용하는 디렉토리
/usr	사용자와 관련된 명령과 범용 프로그램 등을 갖는다
/usr/bin	/bin 에 있는 명령 파일보다 적게 사용되는 명령 파일을 갖는다
/usr/lib	사용자들이 프로그래밍할 때 사용하는 프로그램의 라이브러리
/usr/include	사용자들이 프로그래밍할 때 사용하는 프로그램의 헤더파일
/usr/man	온라인 도움말 파일을 갖는다
/home	각 사용자의 홈 디렉토리

리눅스 파일시스템 (cont'd)



Shell

- 셸 (Shell)
 - 명령어 해석기(Command interpreter)
 - 사용자가 입력하는 명령을 읽고 해석하는 프로그램
- 프로그래밍 언어
 - 셸이 해석할 수 있는 스크립트라는 프로그램을 작성
 - 유닉스를 사용하는데 있어 주요한 인터페이스
- 종류
 - bash, csh, ksh, sh, tcsh

각 Shell 의 특징

- bash
 - Bourne again shell
 - 리눅스에서 가장 많이 사용하는 셸
 - 명령행 편집, 히스토리 치환, 본 셸 호환
- csh
 - C shell
 - 본 셸과 호환되지만 전혀 다른 프로그래밍 인터페이스
 - 명령행 편집 기능 제공하지 않음
- ksh
 - Korn shell
 - 유닉스에서 가장 많이 사용하고 있는 셸
 - 본 셸과 호환, 명령행 편집 기능 제공
- sh
 - Bourn shell
 - Original shell
 - 명령행 편집기능 제공하지 않음
- tcsh
 - 확장형 C shell

Shell 확인 및 변경

- 현재 사용하고 있는 shell 확인

- `$ echo $SHELL`

- ex)

```
$ echo $SHELL  
/bin/bash
```

- 다른 shell 로 바꾸기

- `$chsh`

- ex)

```
$ chsh  
Changing shell for root.  
New shell [/bin/bash] : /bin/csh  
Shell changed.  
&
```

Shell 기본 명령어

- 파일과 디렉토리 목록보기 : ls
 - 숨긴 파일 표시하기 (ex : ls -a)
 - 자세한 정보 표시하기 (ex : ls -l)
 - 하위 디렉토리의 내용 모두 표시 (ex : ls -R)
- 현재 작업 디렉토리 표시하기 : pwd

```
[tonxoo@kde example]$ pwd  
/home/tonxoo/example  
[tonxoo@kde example]$
```

- 디렉토리 이동하기 : cd
 - 절대경로 이동 (ex : cd /home/tonxoo/test)
 - 상대경로 이동 (ex : cd ..)

Shell 기본 명령어 (cont'd)

- 디렉토리 만들기 : mkdir
\$ mkdir [생성 디렉토리명]
- 디렉토리 삭제하기 : rmdir
\$ rmdir [삭제 디렉토리명]
- 파일 삭제하기 : rm
\$ rm [삭제 파일명]
 - r 옵션으로 최하위 디렉토리까지 삭제 가능
\$ rm -rf [삭제 디렉토리명]

Shell 기본 명령어 (cont'd)

- 파일 복사하기 : cp

- \$ cp [원본 파일 이름] [복사 파일 이름]

- \$ cp [원본 파일 이름] [복사할 디렉토리 이름]

- \$ cp [원본 파일 이름] [복사할 디렉토리 이름/복사 파일 이름]

- \$ cp [디렉토리/원본 파일 이름] [복사할 디렉토리 이름/복사 파일 이름]

- \$ cp -r [원본 디렉토리 이름] [복사할 디렉토리 이름]

- 파일이나 디렉토리 이동하기 : mv

- \$ mv [원본 파일] [이동할 디렉토리 명]

- \$ mv [원본 파일] [이동할 파일 명]

- \$ mv [원본 디렉토리] [이동할 디렉토리 명]

- mv 명령어로 파일 또는 디렉토리명 변경 가능

- \$ mv [원본파일 or 디렉토리] [변경할파일 or 디렉토리]

고급 Shell 명령어

- Redirection 사용하기 (>, >>, <)
 - 입출력 사이의 개체 변경 시 사용
 - 명령어 > 파일 이름
 - 명령어의 결과를 파일로 저장(파일존재 시 지우고 저장)
 - 명령어 >> 파일 이름
 - 명령어의 결과를 파일로 저장(파일존재 시 맨 뒤에 첨부)
 - 명령어 < 파일 이름
 - 명령어의 입력으로 파일을 지정

```
[root@localhost home]# echo hello~  
hello~  
[root@localhost home]#
```

```
[root@localhost home]# echo world > test.txt
```

```
[root@localhost home]# echo wow >> test.txt
```

```
[root@localhost home]# cat < test.txt
```

고급 Shell 명령어 (cont'd)

- Pipe 사용하기 (|)
 - 명령어1 | 명령어2
 - 명령어의 출력을 다른 명령어의 입력으로 전달

```
[root@localhost home]# ls | wc -w
```

- 와일드카드 사용하기 (*,?)
 - * : 모든 문자를 대신함
 - ? : 한 문자를 대신함

```
[root@localhost home]# ls -al *.txt
```

- 명령어 순차 실행 (;)
 - 여러 개의 명령어를 세미콜론(;)으로 구분하면 왼쪽에서 오른쪽 순서대로 실행

```
[root@localhost home]# ls -al *.txt >> list.txt; cat < list.txt; ls -al
```

고급 Shell 명령어 (cont'd)

- 백그라운드 실행하기 (&)
 - 명령어 &
 - Shell의 제어를 받지 않고 무한으로 구동시키는 방법

```
[2012504297@localhost ~]$ vi
```

Foreground 실행

```
[2012504297@localhost ~]$ vi&  
[1] 17242  
[2012504297@localhost ~]$ jobs  
[1]+  Stopped                  vim  
[2012504297@localhost ~]$
```

background 실행

Q & A

- Thank you :)