Linux Environment
Hyungsoo Jung  
Distributed Computing Systems Lab.
Logging into Linux Systems

- Text-based (TTY) terminals
  - When you connect to a UNIX computer remotely (using telnet) or when you log in locally using a text-only terminal, you will see the prompt:
    login:
    password:
  - To log out of a text-based UNIX shell
    - Type “exit” or ctrl-d
Hyungsoo Jung
Distributed Computing Systems Lab.
Hyungsoo Jung
Distributed Computing Systems Lab.

login as: comp-ta
Password:
Linux martin 2.4.29 #1 SMP 2005. 01. 22. (토) 00:16:18 KST i686 GNU/Linux
******************************************************************************************
* 이 서버는 동문기업 (주)넥슨(NEXON)에서
* 서울대 컴퓨터 공학부 학생들을 위해 기증한 서버입니다. *
******************************************************************************************

martini 에 오신 것을 환영합니다.

[시스템 사양]
Intel Pentium III - Xeon 700Mhz * 4
Debian Linux 3.1 Stable (Sarge)
Kernel 2.4.x with SMP

[암호 변경]
통합계정을 사용하시는 분은 커뮤니티(http://www.snu.ac.kr)
의 암호 변경 페이지를,
또는 유저의 경우(수업을 위해 발급된 계정)
local-passwd 명령을 사용하여 주십시오.

[Contact US]
시스템 사용에 문제가 있을 경우 이하 주소로 연락해 주세요
Telephone : 02-880-8178
Homepage : http://bacchus.snu.ac.kr
E-mail : contact at bacchus.snu.ac.kr

Last login: Sun Mar 11 21:47:10 2007 from karma.snu.ac.kr
comp-ta@martini:~$
Changing your passwd

- The Linux command to change your password is passwd:
  - $ passwd
  - The system will prompt you for your old password, then for your new password.

- Remember the following points when choosing your password:
  - Avoid characters which might not appear on all keyboards, e.g. '£'.
  - The weakest link in most computer security is user passwords so keep your password a secret, don't write it down and don't tell it to anyone else. Also avoid dictionary words or words related to your personal details (e.g. your boyfriend or girlfriend's name or your login).
  - Make it at least 7 or 8 characters long and try to use a mix of letters, numbers and punctuation.
Exercise 1

$ echo hello world     $ last
$ date                  $ finger
$ hostname             $ w
$ uname -a              $ top
$ ls -al                $ dmesg | more
$ who am i              $ echo $SHELL
$ id                    $ man ls
$ echo 5+4 | bc -l       $ man who
$ yes you are so smart  $ cal 2009
$ history               $ history
The Linux Filesystem
Objectives

- The Linux filesystem and directory structure.
- File and directory handling commands.
- How to make symbolic and hard links.
- How wildcard filename expansion works.
The Linux Filesystem

- Every item stored in a UNIX filesystem belongs to one of four types:
  - **Ordinary files**
    - Ordinary files can contain text, data, or program information.
  - **Directories**
    - Directories are containers or folders that hold files, and other directories.
  - **Devices**
    - To provide applications with easy access to hardware devices, UNIX allows them to be used in much the same way as ordinary files.
  - **Links**
    - A link is a pointer to another file.
    - hard-link, soft-link
Typical Linux Directory Structure

```
/  
  
  bin  sbin  home  tmp  lib  usr  var
  
  jane  will  zeb
  
  work  play
```

Hyungsoo Jung
Distributed Computing Systems Lab.
# Directory & Typical Contents

<table>
<thead>
<tr>
<th>Directory</th>
<th>Typical Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>The &quot;root&quot; directory</td>
</tr>
<tr>
<td>/bin</td>
<td>Essential low-level system utilities</td>
</tr>
<tr>
<td>/usr/bin</td>
<td>Higher-level system utilities and application programs</td>
</tr>
<tr>
<td>/sbin</td>
<td>Superuser system utilities [for performing system administration tasks]</td>
</tr>
<tr>
<td>/lib</td>
<td>Program libraries [collections of system calls that can be included in programs by a compiler] for low-level system utilities</td>
</tr>
<tr>
<td>/usr/lib</td>
<td>Program libraries for higher-level user programs</td>
</tr>
<tr>
<td>/tmp</td>
<td>Temporary file storage space [can be used by any user]</td>
</tr>
<tr>
<td>/home or /homes</td>
<td>User home directories containing personal file space for each user. Each directory is named after the login of the user.</td>
</tr>
<tr>
<td>/etc</td>
<td>UNIX system configuration and information files</td>
</tr>
<tr>
<td>/dev</td>
<td>Hardware devices</td>
</tr>
<tr>
<td>/proc</td>
<td>&amp; pseudo-filesystem which is used as an interface to the kernel. Includes a sub-directory for each active program (or process).</td>
</tr>
</tbody>
</table>
Directory and File handling Commands

- `pwd` (print [current] working directory)
- `ls` (list directory)
  - `ls -al`
Directory and File handling Commands

- cd (change [current working] directory)
  - $ cd /usr/local/
- mkdir (make directory)
  - $ mkdir tmp_dir
- rmdir (remove directory)
  - $ rmdir tmp_dir
- cp (copy)
  - $ cp source-files destination
Directory and File handling Commands

- **mv** (move/rename)
  - `$ mv source destination`
  - If `destination` is an existing directory, the new name for `source` (whether it be a file or a directory) will be `destination/source`.

- **rm** (remove/delete)
  - `$ rm target-file`
  - `$ rm -rf / home/will`
Directory and File handling Commands

- **cat (catenate/type)**
  - `$ cat target-file`
  - displays the contents of `target-file(s)` on the screen, one after the other.

- **more and less (catenate with pause)**
  - `$ more target-file`
Making Hard and Soft (Symbolic) Links

- Direct (hard) and indirect (soft or symbolic) links from one file or directory to another can be created using the `ln` command.

  - `ln –s filename linkname`
  - `ln –s hello.txt bye.txt`
  - `ls –l bye.txt`

```
lrwxrwxrwx 1 will finance 13 bye.txt -> hello.txt
```
File and Directory Permission

- chmod (change [file or directory] mode)
  - $ chmod options files
  - $ chmod 600 private.txt
  - sets the permissions on private.txt to rw------- (i.e. only the owner can read and write to the file).
Inspecting Files

- **file filename(s)**
  - file analyzes a file's contents for you and reports a high-level description of what type of file it appears to be:

- **head, tail filename**
  - head and tail display the first and last few lines in a file respectively.
  - $ tail -20 messages.txt
  - $ head -5 messages.txt
Inspecting Files

- objdump options binaryfile
  - objdump can be used to disassemble binary files
Finding Files

- **find**
  - $ find directory -name targetfile -print
  - $ find /home -name "*.txt" -print 2>/dev/null
  - will search all user directories for any file ending in "*.txt" and output any matching files

- **which command**
  - $ which ls
Finding Text in Files

grep (General Regular Expression Print)

- grep \textit{options \textit{pattern files}}
- grep hello *.txt
  - searches all text files in the current directory for lines containing "hello".
- grep -vi hello *.txt
  - searches all text files in the current directory for lines that do not contain any form of the word hello (e.g. Hello, HELLO, or hELlO).

$\texttt{grep}\ \texttt{hello}\ \texttt{`find\ .\ -name\ "*.txt"\ -print`}$

- will search all text files in the directory tree rooted at the current directory for lines containing the word "hello".
File Compression

- `tar` (tape archiver)
  - To create a disk file tar archive, use
    - `$ tar -cvf archivename filenames`
  - To restore files from a tar archive, use
    - `$ tar -xvf archivename`