ITB721/ITN721

Unix Network Administration

Lecture 2

User and System Management

Lecture Content

- User Account Management
- Account Administration
- Resource Management

User Account Management

- Includes
  - Account creation
  - Account suspension
  - Account deletion
  - Password maintenance

Account Creation Principles

- New accounts must always be authorized
- Administrators should have a checklist
- Account rights must be appropriate for new account
- Account provision on successful identification

Account Suspension Principles

- Account suspension
  - prevents access; and
  - preserves the account
- Account suspension - first stage of departing employee process
- Often other circumstances when account suspension may be necessary

Account Deletion Principles

- Account deletion - typically some time after employee has left organization
- Can be dangerous to immediately delete account
  - Common to initially suspend/disable account

User Authentication

- Verifying a claimed identity
- Common two step authentication process in computer systems
  - supply claimed identity (eg username or login name)
  - provide authentication token (eg password)

Good -v- Bad Passwords

- Not associated with user
- Not dictionary word/s
- Include mix of upper case, lower case, numbers and symbols
- Changed regularly
- Unique to one system

- Names
- Common words
- Associated with user
- Any of the above in reverse, or with numbers only mixed in
- All lower/upper case
- Written down somewhere

Expiring and Changing Passwords

- Passwords should expire regularly
  - but not too frequently
- Need to reach suitable balance for expiration frequency
  - Typically, 30-60 days is common
- New password should significantly differ from old password

Lost and Forgotten Passwords

- Users forget passwords
- Initial password may be lost
- Administrator needs to reset or re-issue passwords
  - Do not do this without proof of identity

Account Administration

- Accounts
- Directories
- Files
- File Security
- Shell Scripts

Accounts

- Accounts on the system created, suspended and deleted by the superuser
- Password needs to be set with each new user account
- Typically no password required for system accounts

User Accounts in Linux

- Each account should have
  - A unique login name
  - A unique numeric user ID
  - A default group
  - A home directory
  - A default shell
- Creating an account adds an entry into /etc/passwd file

Format of /etc/passwd

```
username:x:uid:gid:full-name:home-directory:default-shell
```

Examples:

```
root:x:0:0:root:/root:/bin/bash

n1234567:x:500:600:John Black:/home/n1234567:/bin/bash

swong:x:501:600:Steven Wong:/home/swong:/bin/bash
```

Shadow Passwords

- Each user’s encrypted password stored in /etc/shadow
  - indicated by "x" in second field of /etc/passwd entry
- /etc/shadow only readable by root user

Numeric User ID

- Security actions and rights can be referenced against numeric user ID
- Normally, lower IDs (eg below 500) reserved for system users
  - Not real users, but special accounts
- Root user has numeric user ID of 0

Groups in Linux

- Groups listed in /etc/group
- New groups added to system using “groupadd” command (or by editing /etc/group)
- Users in group/s other than their default group, listed in /etc/group
- Special groups exist for specific purposes

Shells

- Provide command line interface and enable basic editing of command line
- Keep track of current directory
- Interpret directory paths
- Shells also provide eg
  - Customizable prompts
  - Command line history

Default Shell

- Program run whenever new terminal window opened, or when user logs in, (or system account invoked)
- One associated with each account
- Often a command line interface shell
- If non-functional
  - command line interface not permitted for account

The "useradd" Command

- "useradd" or "adduser" automates the process of adding user accounts
- All required options supplied on "useradd" or "adduser" command line
- Also possible to add users manually

Suspending User Accounts in Linux

- Common technique - lock account by placing "*" at start of password field in /etc/passwd
  - prevents user from logging in, but can be easily reversed
- Can also eg change default shell

Deleting User Accounts in Linux

- To entirely delete a user account, use "userdel" command with appropriate option
  - Caution: not recommended as a first step
  - This irrevocably removes the user

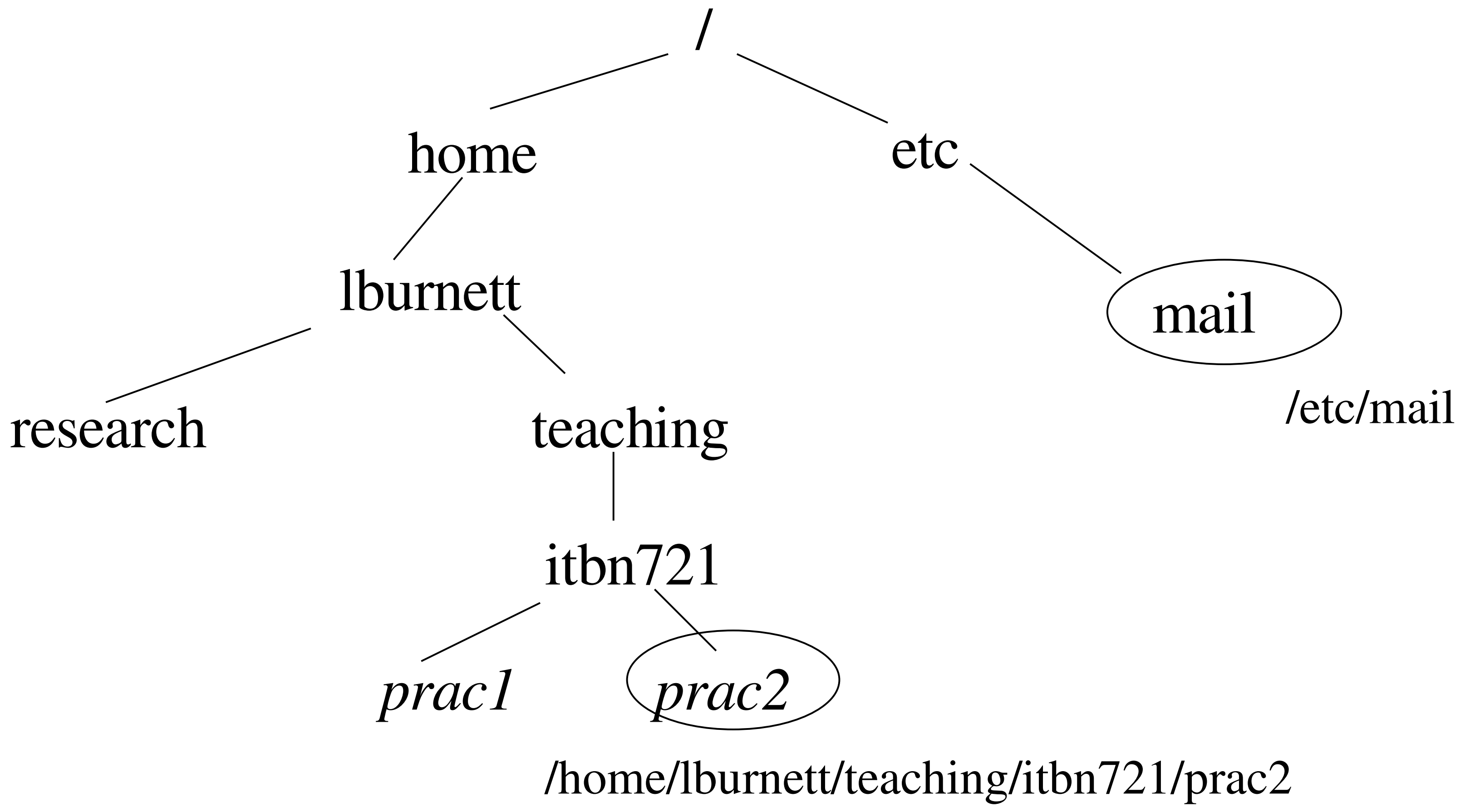
Directories

- Hierarchical structure
- Top level is the root directory (written with a single slash, "/")
- Directories contain files and other directories
- Possible to nest levels of directories
- When written in a path, directories are separated by a "/"

Absolute Paths

- Written with a leading "/"
- Interpreted as starting from the root directory
- Allows access to a file or directory which is not found within the current directory path (subject to appropriate permissions)

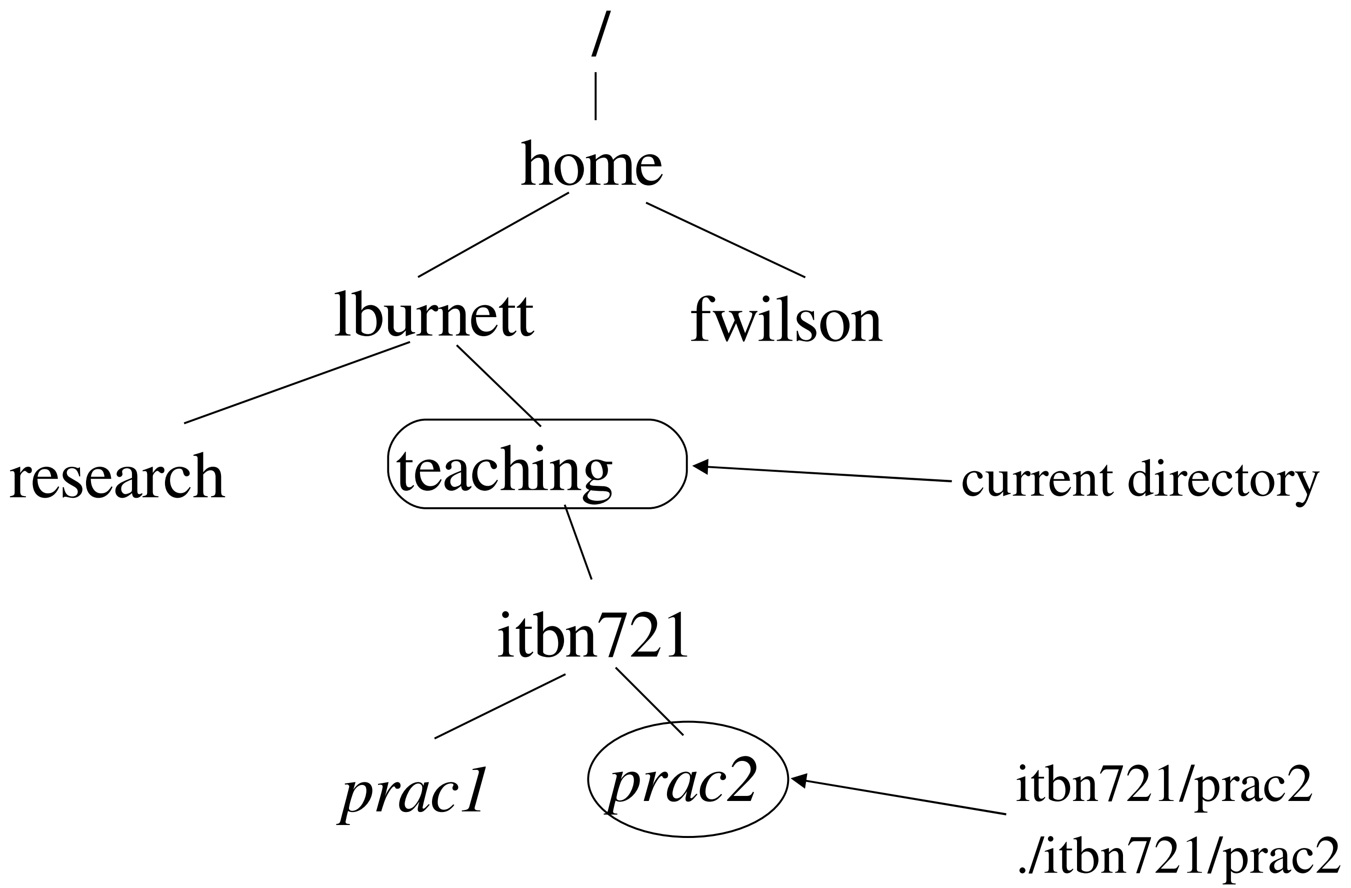
Example: Absolute Path



Relative Paths

- Written without a leading "/"
  - Can also be written with a leading "./"
  - The "." refers to the current directory
- Interpreted as starting from the directory you are currently in
- A ".." refers to the parent directory

Example: Relative Path



Home Directories

- One specified for each user
- Tilde "~" represents the home directory of the currently logged in user
  - Equivalent to specifying /home/<username>
- Only accessible (by default) by logged in user, and root user

Standard Directories

- Certain directories are used for particular files in Linux, for example,
  - /bin and /usr/bin store user executable programs
  - /sbin and /usr/sbin store system executable programs
  - /tmp contains temporary files
  - /etc contains configuration files
  - /var contains log files, spool files, etc

Files

- Types include
  - Ordinary or regular eg configuration files, source code, executable files, library files
  - Special
  - Directories
  - Symbolic links

Filenames in Linux

- Maximum 255 characters in length
- Made up of letters, numbers and some symbols
- Certain symbols have special meaning to the shell and cannot be used

Device Files

- Generally stored in /dev directory
- Allow interaction between hardware devices and file system
- Pseudo devices do not refer to any hardware
  - eg /dev/null, /dev/zero

Block and Character Devices

- Block devices read and write blocks of data at a time
  - eg hard disks
  - eg device files - /dev/sda1, /dev/sda2
- Character devices read and write one character at a time
  - eg printers, terminals
  - eg device files - /dev/usb/lp0, /dev/tty2 respectively

Symbolic Links

- Used to point one file (directory) to another file (directory)
- Enable the same file (directory) to be accessed from different locations on the system
- Removing a symbolic link does not delete the file (directory)

File Security

- Aspects include
  - Assigning file and directory ownership
  - Setting file and directory permissions
  - Integrity checking (discussed in a later lecture)

Superuser and Files

- Superuser has control over all aspects of system, for example
  - Can read any file
  - Can change any file
  - Can delete any file
  - Can change ownership and/or permissions
- Normal users may only control files they own

File Ownership

- In Unix, every file and directory has an owning user and an owning group
- Default for new files/directories is the user and group of the creator
- No ownership provisions for users other than owning user and owning group
- Changes accomplished using "chown" and "chgrp" commands

Users and Groups

- Each user must belong to at least one group
- Groups are useful for setting permissions for a group of users
- File permissions can be set for individual users, and groups of users, owning a file

File Permissions

- Permissions are attached to all files and directories in a Unix system
- Permissions describe what type of access users, or groups of users, have to a file
- Can be changed by owning user, or root user

Unix File Permissions

- Read permissions
- Write permissions
- Execute permissions
- SetUID/SetGID
  - Executes with rights of owner/owning group, instead of executor
- Sticky Bit
  - Only owner of file, owner of directory, or root user can delete files within

Unix File Permissions

- Owner
  - Read, Write, Execute, SetUID
- Group
  - Read, Write, Execute, SetGID
- Other *(users who are not the owning user and are not in the owning group)*
  - Read, Write, Execute

Listing File Permissions

- "ls -l" command line lists file permissions etc
- Format of listing includes

  type   owner group other

  d/c/b/l   rwx   rwx   rwx
- rwx for each of owner, group and other refer to read, write and execute
- Types of files include
  - d: directory, c: character device, b: block device, l: symbolic link

Some Special Permissions

- If owner field has "s" in place of "x", then SetUID executable
- If group field has "s" in place of "x", then SetGID executable
- If other field has "t" in place of "x", then sticky

Octal Representation of Permissions

- Permissions can be written as 4 octal digits
- First digit in binary:
  - SetUID SetGID Sticky
- Remaining 3 digits in binary correspond to Owner, Group and Other permissions
  - Within each set, in binary:
    - read write execute

Octal Permissions: Examples

- 0640 = 000 110 100 000 (rw- r-- ---)
  - Owner: Read Write
  - Group: Read
  - Other: None
- 4755 = 100 111 101 101 (rws r-x r-x)
  - Owner: Read Write Execute SetUID
  - Group: Read Execute
  - Other: Read Execute

Changing Permissions

- Use "chmod" command
- Can be specified either in octal (numeric) mode or in text (symbolic) mode
- For example, the following are equivalent:
  - “chmod 0755 <filename>”
  - “chmod u=rwx,g=rx,o=rx <filename>”

Special Files and their Permissions

Examples of Device Files

- Hard disk drive
  - `brw-r-----  root   disk   8,2          /dev/sda2`
- USB flash drive
  - `brw-r-----  root   disk   8,17         /dev/sdb1`
- Printer
  - `crw-rw----   root   lp     6,0          /dev/usb/lp0`

Symbolic Links and their Permissions

Example of Symbolic Link

- lrwxrwxrwx root root /dev/dvd -> /dev/hda
- /dev/hda can also be referred to as /dev/dvd
- Symbolic links always have permissions 777
- Actual permissions are that of real file (/dev/hda in this case)

Shell Scripts

- Enable partial automation of commands
- Used to execute a pre-programmed sequence of commands
- Text file containing the commands
- Often used in configuration management
- When script is run, it runs with permissions and rights of executing user
- SetUID/SetGID script will run with rights of owner/owning group

Writing Linux Shell Scripts

- Use a text editor
- First line of shell script specifies which shell to run script in eg #!/bin/bash
- Subsequent lines are command lines listed sequentially
- Manual page for the shell (eg “man bash”) describes shell commands, parameters and variables usable within the shell
- Shell scripts may contain control structures (Compound Commands), eg if ... then ... else, for loops, while loops

Shell Variables and Parameters

- Some example variables include
  - HOME=/home/jblack
  - PATH=/bin:/usr/bin
  - PS1='$ '
- Some example command line parameters to shell scripts include
  - $1 $2 $3 ... (positional parameters)
  - $#

  total number of command line parameters entered (special parameter)

Executing Shell Scripts

- If executable, type in name of script, if in $PATH
  - If not in path, type “./<scriptname>”
- If not executable, use eg "sh" command

Redirection of Output

- Any executable, script or otherwise, can have output redirected from screen to a file or to another program
- ">" symbol redirects output to named file
  - Eg “ls > outfile”
- "|" symbol redirects output to be input to next command
  - Eg “ls | wc”

Redirection of Input

- A program that needs input from keyboard can read such input from a file
- "<" symbol redirects named file to become input to program

Regular Expressions

- Used to specify patterns and wildcards
- Used by many commands eg grep, find
- Described in the man page for “grep”. Examples:

– .	matches any character
lecture..	The word "lecture" followed by any 2 characters
– ^	start of line
^The	A line starting with the word "The"
– $	end of line
end$	A line ending with the word "end"

Resource Management

- Includes
  - Distribution of resources
  - Accounting for resources
  - Installing and updating software
  - Managing software licences

Resources

- Hardware resources include servers, workstations, network devices and equipment
- Software resources include operating systems, applications, packages
- Must ensure legal use

Software Installation

- Software packages can provide additional functionality to system
- Patches and bug fixes often distributed as packages
- Package managers (eg Red Hat Package Manager) often used to install, delete, query, verify (and generally manage) packages
- Automated package managers also available

Red Hat Linux / Fedora Core Distribution Packages

- Red Hat Linux and Fedora Core use RPM (Red Hat Package Manager)
  - "rpm" command (with various options) used
  - .rpm for binary executable packages
- Contain files to be installed into identified directories
- Contain an information block, includes
  - name, version number, size, summary, dependencies, author

Package Considerations

- Which package to use?
  - Often, multiple versions of same package
  - Necessary to check that correct package is used
    - Eg Kernel version, Linux distribution, library (glibc) version
- Package Dependencies
  - All dependencies must be installed before desired package can be installed
  - Dependencies can insist on a particular version number, or a minimum version number

Red Hat Package Manager (rpm) Command Lines

- Install
  - # rpm -i<options> <package>
- Update
  - # rpm -U<options> <package>
- Query
  - # rpm -q<options> <package>

Non-Distribution Packages

- Sometimes, packages distributed as "Tape Archive" files, .tar and .tar.gz (or .tgz)
- "tar" command used to extract relevant files for installation
- Install scripts normally provided in package archive to automatically install packaged files into correct locations

Source Distributions

- Sometimes, source is distributed for user to self-compile
  - .srpm for source code packages
  - Also distributed as .tar or .tgz files
- To unpack / extract
  - Use “rpm” or untar appropriately

Compiling and Installing Source Distributions

- Often necessary to configure compile options first
- Common to have a script probe the system and ask user relevant questions
- Compilation usually through "make all" command line
- Once compiled, compiled executables and other supporting files must be moved (installed) to correct locations
  - Generally, "make install" will achieve this

Licence Management

- Ensure sufficient software licences are purchased
- Be aware of terms of shareware licensing, which may include
  - Commercial use limitations
  - Time usage limitations
- Some products have free licences

Common Administrator Tasks

- Creating, suspending and deleting user accounts
- Resetting passwords for users
- Granting users and groups permissions to appropriate directories
- Setting up shared directories for users
- Managing resources

General Ethical Considerations

Topics relevant to this lecture include

- Account Management
- File Management
- Resource Management
- Licence Management