USER, ACCESS CONTROL AND PROCESSES

UNIX Programming 2014 Fall by Euiseong Seo

Access Control

- Access control determines who can do what
  - Can user ‘A’ log into a system ‘X’?
  - Can user ‘A’ delete file ‘P’ on filesystem?
  - Can user ‘A’ reboot the system?
- Raison d’etre of an operating system
- Access control is closely tied together with security

General Rules of Access Control

- Objects (e.g., files and processes) have owners
- Owners have broad (but not necessarily unrestricted) control over their objects
- You own new objects that you create
- The special user account called “root” can act as the owner of any object
- Only root can perform certain sensitive administrative operations
  - Change the system time
  - Format a new hard disk
  - And so on

Users (or Accounts)

- A user is identified by UID (user ID)
- User list is stored in /etc/passwd
  - Each line represents one user
  - Each line contains seven fields separated by colons
    - Login name
    - Encrypted password placeholder
    - UID
    - Default GID
    - GECOS information (full name, office and so on)
    - Home directory
    - Login shell

```
euiseong:x:1000:1000:Euiseong Seo,,,:/home/euiseong:/bin/bash
```

Users (or Accounts)

- Login names
  - (Traditionally) 8 alphanumeric characters
  - Must be unique within a system
- Encrypted password
  - Old systems put encrypted passwords in the 2nd field
    - Empty value means “no password required”
  - Modern systems put passwords separately in /etc/shadow

```
syslog:*:15412:0:99999:7:::
euiseong:$6$Q9PX1Z4k3w.qF8C977tZh/rFLs/N2TbTesnEwcSzrnQb0/znYXTCny.heBtkeDpVDFPh
XnKw75ioYzzJvSxbKlLQ1r0:15412:0:99999:7:::
```

    - Can set expiration date, password change period and so on
  - Encryption is done through crypt function of C library

Users (or Accounts)

- UID number
  - Login names are provided for the convenience of users
  - Software and filesystem use UIDs internally
  - A UID is a 32 bit integer
  - UID 0 is reserved for the root user
  - Multiple login names have the same UID
    - Do not recycle UIDs
- GID number
  - A GID is a 32 bit integer, too
  - Defines default group for a user
  - Default group for a user is only one while a user can belong to multiple groups

Users (or Accounts)

- GECOS field
  - Record personal information
  - finger command interprets comma-separated entries in the following order
    - Full name
    - Office number and building
    - Office telephone extension
    - Home phone number
  - These can be set by using chfn command

Users (or Accounts)

- Home directory
- Login shell
  - Path of default login shell
  - Set by using chsh command
    - Linux limits changes to shells listed in the file /etc/shells

Pseudo Users

- Some files are not user-specific, but need not to be owned by root
  - These files are owned by appropriate pseudo users
- Pseudo users are regular entries of /etc/passwd, but not allowed to log in
  - Their login shells are set to be /bin/false or /bin/nologin

```
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

Groups

- A user can belong to multiple groups
- Groups and their members are defined in /etc/group

```
sys:x:3:
adm:x:4:syslog,euiseong
```

- Groups are necessary for file sharing among a group of users
  - Every file has two owners, user owner and group owner
  - E.g.) /var/www/index.html is owned by group www-data
- Default group of a user determines group owner of new files that are created by the user

Access Privileges of a File

- Access privileges of a file determines who can do what to the file
- Who - user owner, group owner and others
- What – read, write and execute
- This information is packed to a 9-character string

```
-rwxrwxr-x 1 euiseong euiseong    8460 Aug 27 15:02 a.out
-rwxrwxr-x 1 euiseong euiseong    8460 Aug 27 15:03 loop
-rw-rw-r-- 1 euiseong euiseong      54 Aug 27 15:03 loop.c
-rw-r--r-- 1 euiseong euiseong  358363 Aug 27 22:08 preliminarystudy.docx
```

- {RWX} {RWX} {RWX} = {User}{Group}{Others}

Root Account

- An omnipotent administrative user
  - Often called ‘super user’
- root’s UID is always 0
- UNIX allows root to perform any valid operations to any files or processes
- Login to root account is generally forbidden for security concerns
  - Then how to use the super power of root?

Process and Its Components

- A process is a program in execution
- Components of kernel data structure for a process
  - Process address space map
  - Current status of process (sleeping, stopped, runnable, etc.)
  - Scheduling priority
  - Information about resource usage
  - Information about opened file descriptors and network sockets
  - Signal mask
  - Owner of process
  - PID
  - PPID (Parent PID)

Process Privilege

- UID
  - UID is the original owner of the process who initiated the process
  - Identify of a process
- EUID (effective UID)
  - Determines what resources and files the process has permission to access at the current moment
  - Permission of a process
  - EUID can be changed dynamically
- Why both UID and EUID are required?

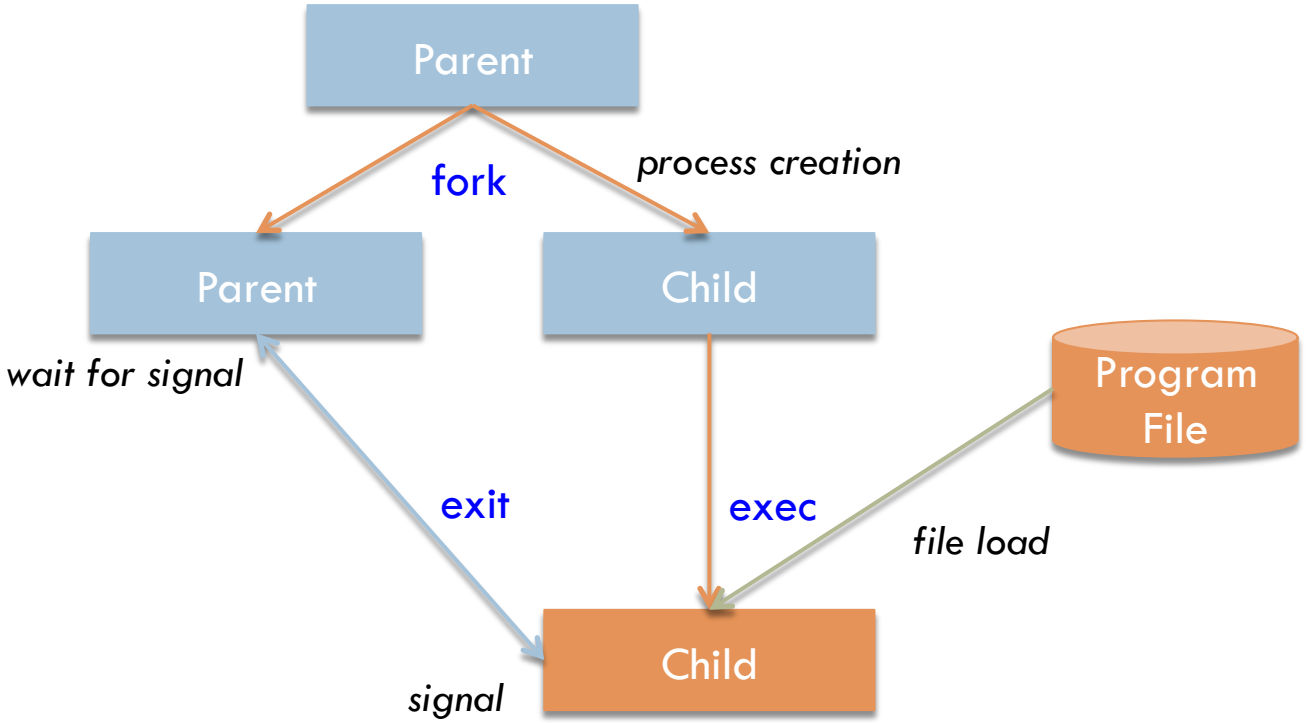
setuid Permission

- /bin/passwd
  - To change the password of a user
  - Any user can change his/her own password
  - How to deal with permission problem?

```
euiseong@accept:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1317 Jun 11 21:38 /etc/passwd
euiseong@accept:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 47032 Feb 17  2014 /usr/bin/passwd
euiseong@accept:~$
```

  - s instead of x means setuid permission
- EUID of passwd process becomes UID of the file owner, root

Fork and Exec Mechanism



Signals

- Process-level interrupt requests
  - Basically, sent by kernel to processes
  - Processes may request the kernel to send signals to other processes
  - Users can send signals via special key combinations
    - CTRL+C for SIGINT
    - CTRL+Z for SIGSTOP
  - Users can send signals via kill command
- About thirty different kinds are defined

Signals

- Representative Signals

#	Name	Description	Behavior
2	SIGINT	Interrupt from keyboard	Terminate
4	SIGILL	Illegal instruction	Terminate
6	SIGABRT	Abort signal from abort function	Terminate and dump
8	SIGFPE	Floating point exception	Terminate
9	SIGKILL	Kill signal	Terminate
11	SIGSEGV	Invalid memory reference	Terminate and dump

- Signals can be caught or ignored
  - By signal handler functions installed in each program
  - Except SIGKILL and SIGSTOP
- Sending signals via kill command
  - kill [-signal#] pid
  - kill -9 -1

Process States

- A process must be in one of the following states
  - Runnable
    - Process can be executed
  - Sleeping
    - Process is waiting for some resource
  - Zombie
    - Process is trying to die
  - Stopped
    - Process is suspended (not allowed to execute)
    - By SIGSTOP (CTRL+Z)
    - Can be continued by SIGCONT

Nice Value

- Runnable processes share processor resource
  - But, not equally distributed
- Priority-based scheduling
  - Important processes have high priorities
  - The higher the priority is,
    the more the processor resource goes
- Nice (process priority in UNIX)
  - -20 to 19 (smaller is higher)
  - 0 is default
  - Normal users cannot raise nice value higher than 0

Nice Value

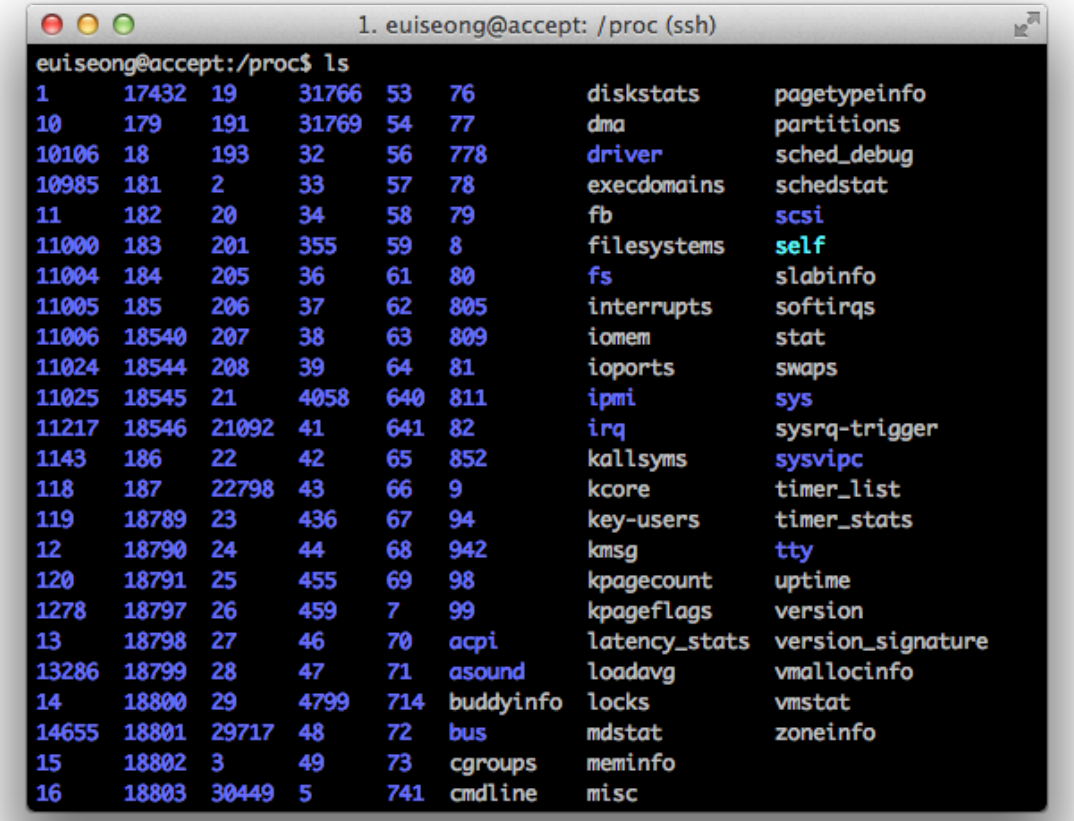
- nice
  - Run a program with modified scheduling priority
  - nice [option] [command]
  - nice -n 5 ~/bin/longtask
- renice
  - Alter priority of running processes
  - renice priority [-p pid] [-u user]
  - renice -5 8829

Monitoring Processes

- ps -aux
- top
  - Dynamic monitoring of running processes

/proc Filesystem

- /proc directory contains useful information



/proc Filesystem

- /proc filesystem is a pseudo-filesystem
  - No actual storage is required and allocated
  - File size is 0, but contents of a file will be generated when you try to read it
- man proc describes information what /proc provides
- /proc/[pid]/ contains files about process [pid]
  - E.g.) /proc/[pid]/io shows I/O statistic of process pid
- Valuable information for system performance profiling or optimization