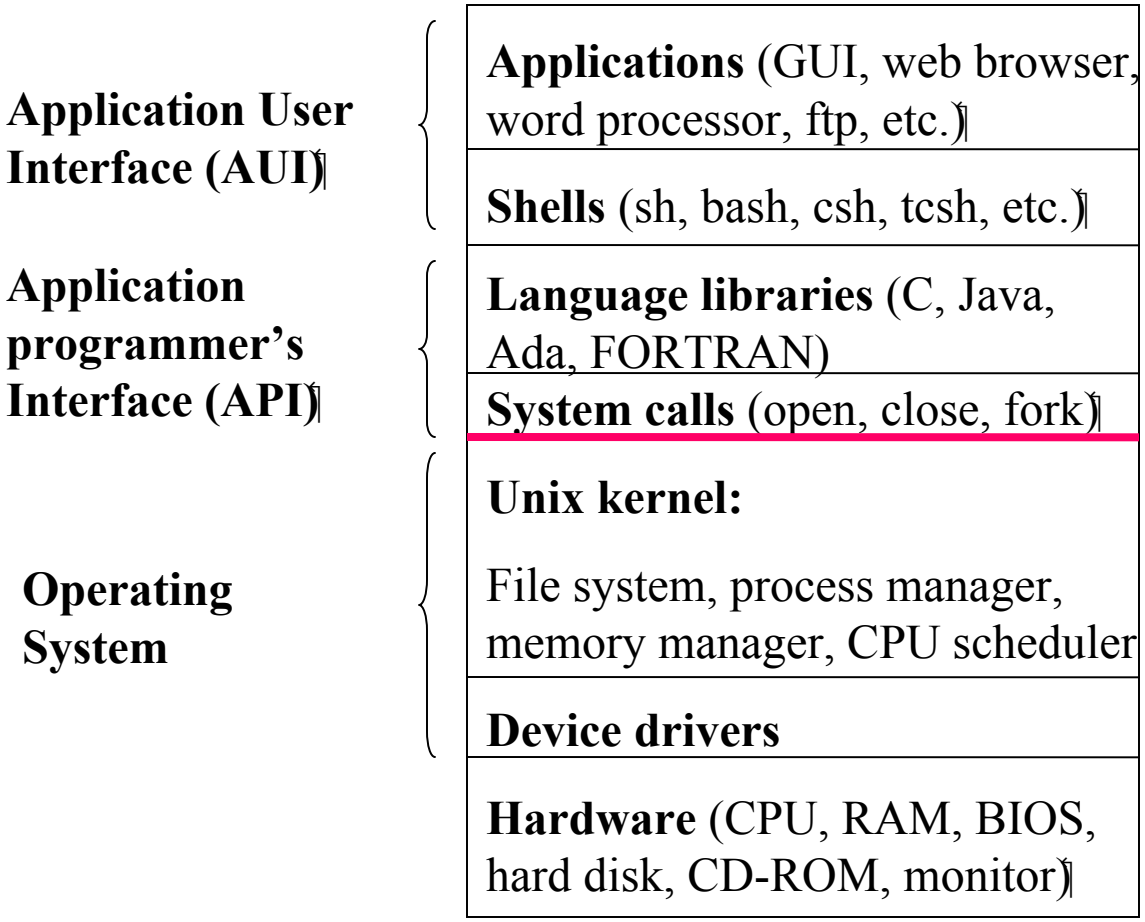
Why Learn Unix (or its variations)?

- Many Unix systems (Solaris on Sun workstations, IRIX on SGI workstations, AIX on IBM servers, various versions of Linux on PCs, FreeBSD, OpenBSD)
- Unix provided the basis (since 1970s) for many operating system concepts and features (multi-tasking, shell and scripting, hierarchical file systems)
- Apple Computer's Mac OS X is Unix-based, see a wikipedia article, and a brief Unix timeline

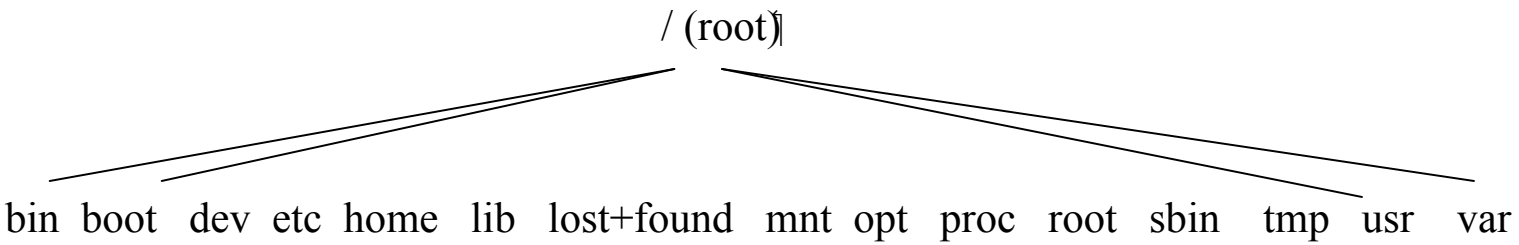
- Mac OS X and Unix articles (tutorial, advanced Unix, how-to's)
- The Law Enforcement and Forensic Examiner's Introduction to Linux, v.3.65 by Barry J. Grundy

Layered Structure of the Unix/Linux Operating System:



Unix file systems:

- The Unix operating system started out as a file system. All system entities are considered as files, including regular files (text or binary), directories, devices, links, pipes, and sockets.
- A typical file system tree looks as follows:



/bin – binary (executable) images of commands such as cat, chmod, cp, kill, ls, mkdir, mv, ps, pwd, rm, rmdir, su, vi

/boot – image of kernel to boot the system

/dev – devices (special files) including character special and block special

/etc – configuration information

/home – user home directories
/lib – language libraries for C, C++, Java, Ada, FORTRAN

/lost+found – files not connected to other directory, which are found using the tool fsck (file system check)

/mnt – mount points for other file systems such CD-ROM, floppy, using the mount command

/opt – add-on packages

/proc – process (task) information

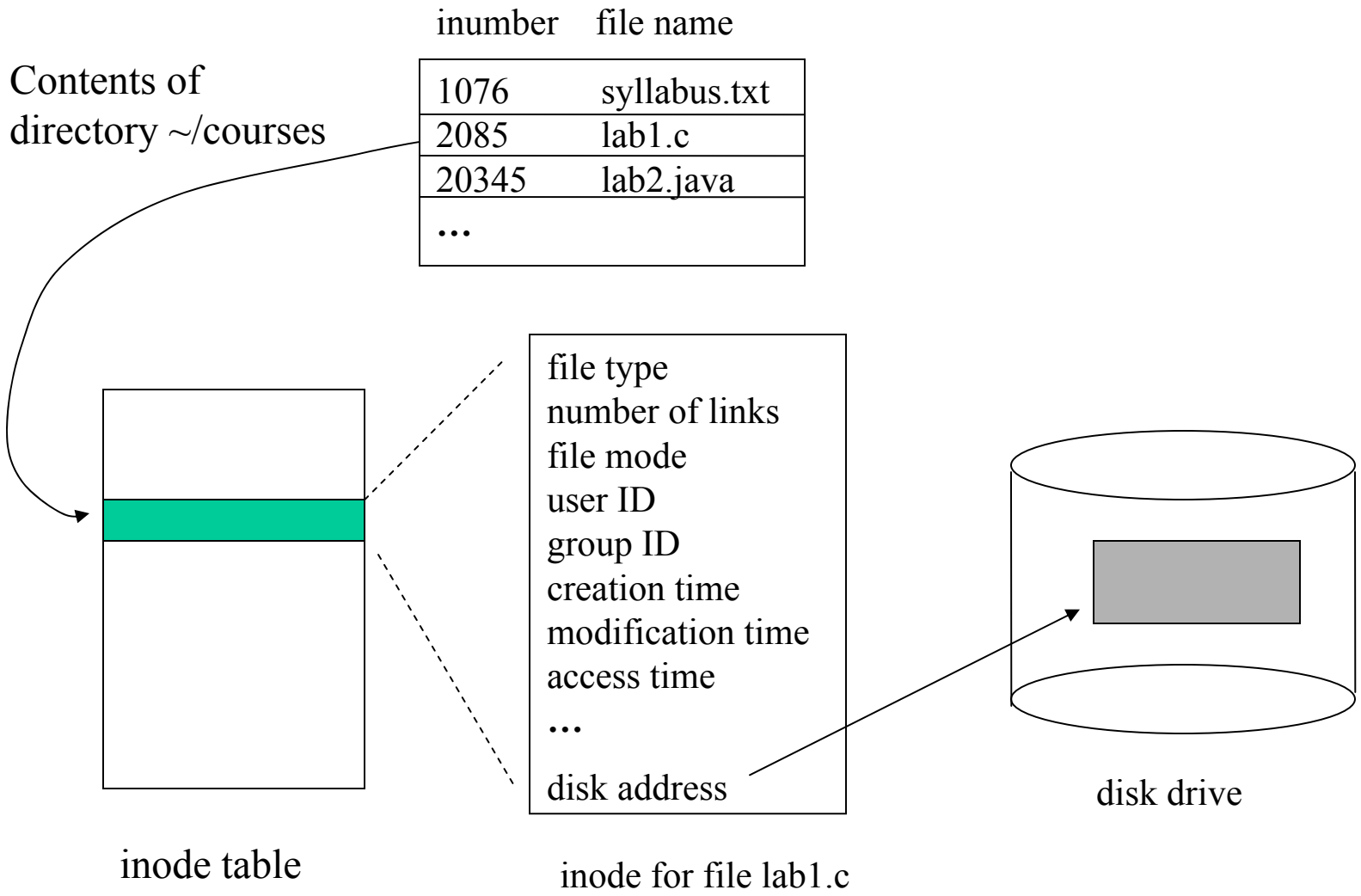
/root – information about the root (administrator) account

/sbin – system administration tools such as init, shutdown

/tmp – temporary files used by several commands (e.g. editor)

/usr – contains subdirectories bin, doc, include, lib, local, man, src, tmp, shared by all users
/var – variable data such as incoming mail

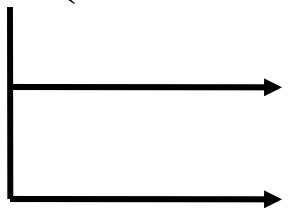
How are files represented and saved on disk:



Common Unix shells and commands:

- shells with increasing functionality:

sh (Bourne shell) → bash (Bourne again shell, used in Linux)



ksh (Korn shell) → zsh

csh (C shell) → tcsh (TC shell)

- File-related commands:

ls (list contents of directory, similar to dir in DOS, options include -l for long format, -a to show all files, -R to show files in the entire directory tree, -t to show last modified file first)

cat (concatenate and display file, e.g. cat lab1.c)

rm (remove file); mv (move or rename file); mkdir (create directory); rmdir (delete directory file); cd (change directory); pwd (show current or working directory); chmod (change protection mode); chown (change owner)

- Process-related commands:

  ps (report process status)

  w or who (display information about logged-in user)

  kill (terminate process)

  top (display and update top CPU users)

  fg (bring background process to foreground)

- editors (vi, pico, emacs)
- special symbols understood by the shell:

  < (input redirection, e.g. a.out < data.txt);
  > (output redirection); >> (append);
  & (run in the background, e.g. a.out&); | (pipe);
  wildcard characters such as * (matches all), ? (matches any)

Unix File Security:

- password-based protection

  Users are identified at login by the user's login name and password; file accesses are then based on the associated user id

- encryption-based protection

  use crypt or des (data encryption standard) to encrypt files

- access-permission based

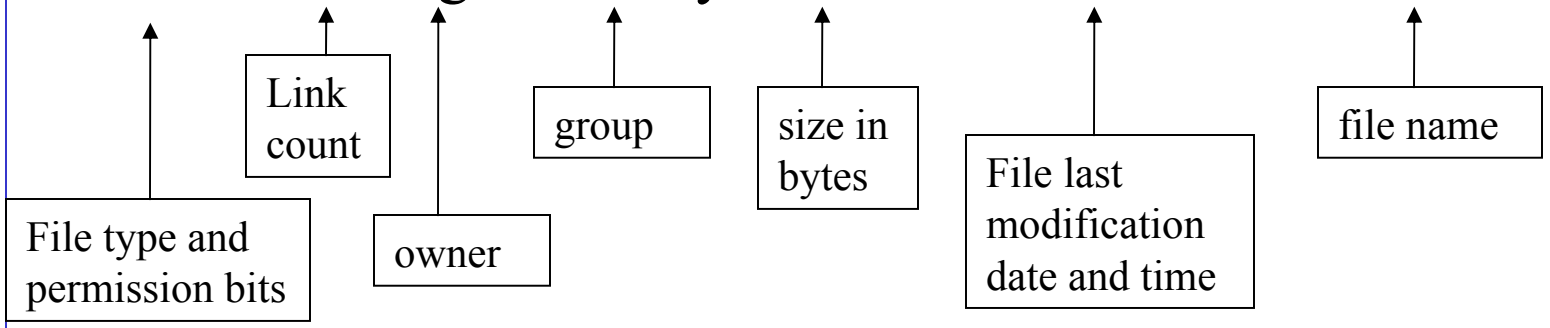
  Each user has an associated UID (user ID) and GID (group ID); the file permission bits prescribe the access rights to 3 kinds of users: the owner, users in the same group, and others; the owner of the file can change the file's permission bits using the chmod command.

Long listing of files and file permissions:

Consider the output of the ls –lt command:

$ ls –lt

```
drwxr-xr-x 2 lang    faculty    512     Jan 15 13:46  sp2002
-rw-r--r--   1 lang    faculty    26624 Jan  5  2001  syllalgfa00.doc
-rw-r--r--   1 lang    faculty    741    Dec 14  2000  score.fa00.out
```



Note that the file permission bits consists of 10 characters, the first indicates file type (d for directory, - for ordinary, b or c for devices, l for symbolic link), the next 9 characters are combinations of r, w, x (and s) in 3 groups.

File access permissions:

r – read access; w – write access; x – execute access (or search for directory files); s – set-user-id turn on (so that when executed the effective user id is temporarily changed to that of the owner's)

Thus, the permission drwxr-xr-x means rwx rights for the owner, r (read) and x (access, search) rights for the group and for others; similarly, the permission -rw-r--r– means r (read) and w (write) rights of the owner, r (read) right for group and others.

Note that the x permission means access or search for directory files, that is, the right to change directory (cd) to there, to access any file located in that directory, etc. The administrator of a Unix system has a user name 'root' who has access rights to all files.

How to use the chmod command:

• use the letters u (user), g (group), o (others), a (all) for the who; use + (to add), - (to remove) and = (to set) for privileges; or

• use octal (base 8) numbers for combinations of the rwx rights (r = 4, w =2, and x =1).

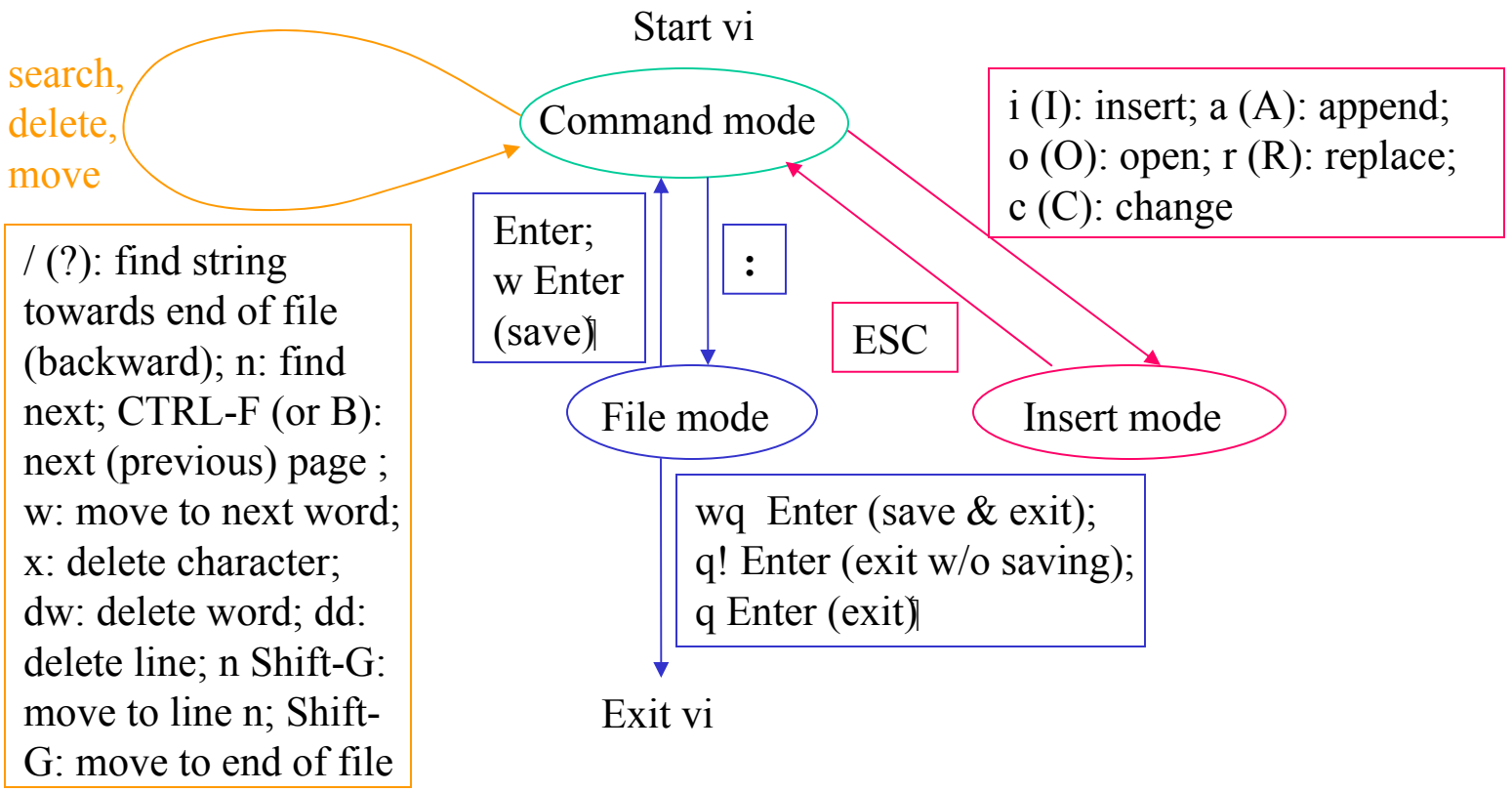
Examples of chmod commands:

chmod 700 * (set rwx rights to user), none to group or others
chmod 644 lab1.c (set rw rights to user, read only to group and to others)
chmod u=rwx courses (set rwx rights to user, and keep the same rights for the group and for others)
chmod go-w lab1.c (remove the write access from group and from others, but everything else remains the same)

Advanced File Processing:

- strings – find ascii strings in any file; use –n length to specify minimum string length (default 4)
- file – determine file type (directory, text, executable, graphics, etc.); check the file /etc/magic for the types of files
- grep or egrep – search file for string pattern; egrep allows full regular expressions to specify patterns (e.g., 'pat1|pat2')
- find – find files with the specified conditions (e.g. name, access time) with the specified directory tree
- whereis – locate binary, source, manual files for a command
- which – locate command (output its full pathname)
- gzip (gunzip) – compress (uncompress files)
- tar – create tape archives, add or extract files from a tar file

- dd – convert and copy file, disk dump (with specified block size, skipped blocks, and type of conversion)
- more, less – browse or page through a file (less is more efficient, and supports editor vi-type commands)
- vi – the ubiquitous visual (key-based) editor:



Evidence of a Compromised File System:

• Stolen passwords: download the password file /etc/passwd and crack passwords (this problem is minimized when a password shadow file /etc/shadow is used which contains the encrypted passwords and is not accessible except by the root); use a network sniffer that intercepts user name/password if they are in clear text form; gain root access by exploiting software bugs (there are many such hacker tools)

```
% ls -l  /etc/passwd
-r--r--r--   1 root      4641701 Jan 18 11:55 passwd
% ls -l  /etc/shadow
-r--------   1 root      2307046 Jan 22 06:17 shadow
% less /etc/passwd
root:x:0:1:Super-User:/:/bin/ksh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
……
slang:x:8563:80:Sheau-dong Lang:/ucf0/pegasus/s/slang:/bin/csh
% less /etc/shadow
Cannot open /etc/shadow
```

Fields (separated by :) of passwd file entries:

Login name
encrypted password (x if shadow file used)
UID
GID
Name, office, phone, etc.
Home directory
login shell

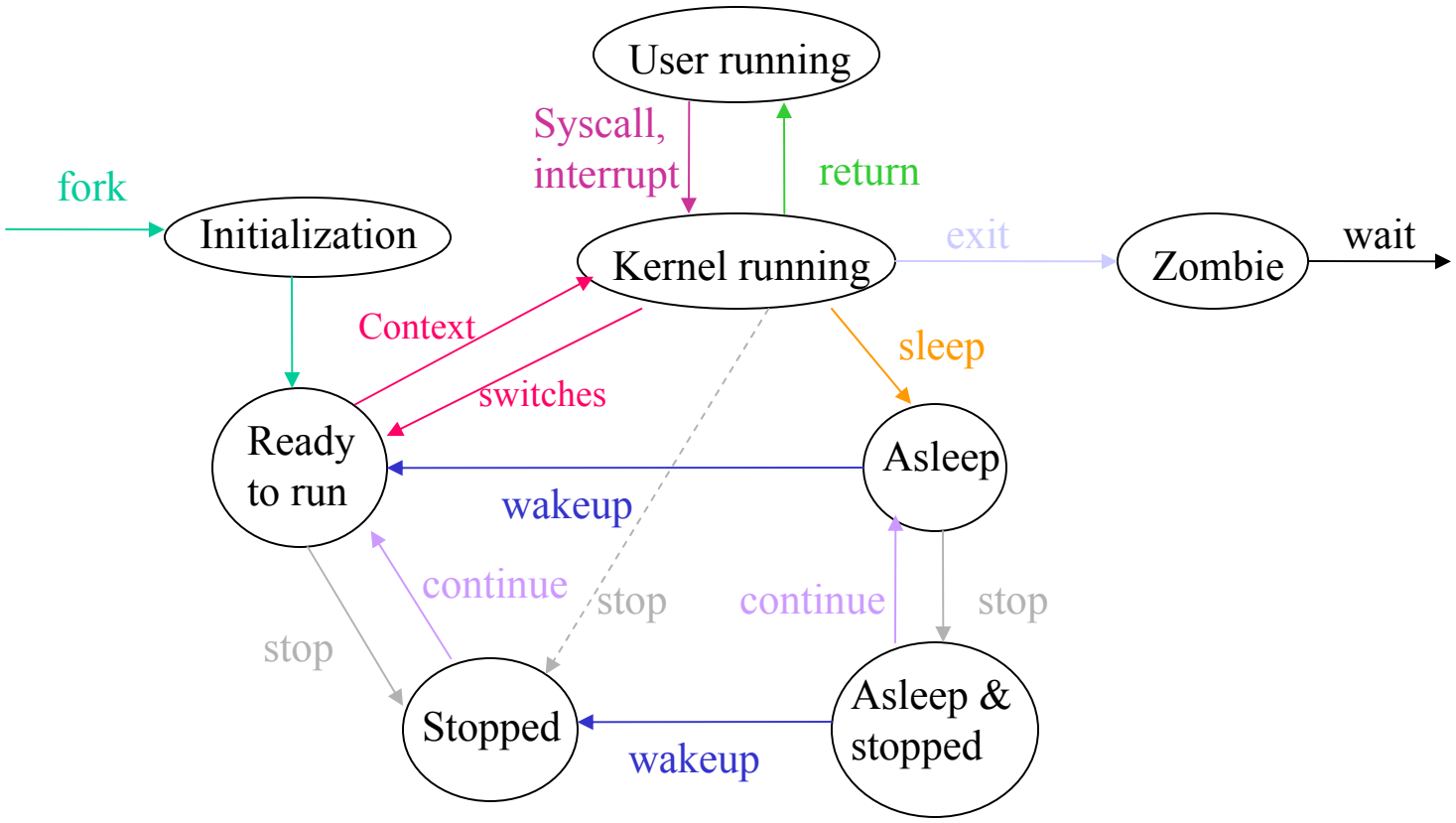
- Suspicious commands in the history file: the history file saves the most recently used commands of the user (e.g. in file ~/.bash_history under the home directory ~).
- Suspicious events in log files: there are various system log files recording events such as login/logout, root access via su or sudo, ftp and tcp connections, and a general system event logger, located under /var/adm or /var/log but have different names on different Unix systems; on Red Hat Linux all log files are under /var/log and are clearly named.
- Useful information can be found in system configuration files located under /etc which shows scheduled processes, location of syslog file, password files, etc.
- Running processes (tasks) are represented as files under directory /proc where evidence of rogue processes can be found.

Unix Processes:

• An operating system runs many applications, such as a shell (command interpreter), commands submitted to the shell including a user program (an executable file), a GUI, applications started from the GUI, etc., each of these running applications constitutes a process (or task). Thus, a process is a "program" in execution, including the code, data areas (static and dynamic), its status, and associated resources.

• Programs are files, so they are protected by the file's permission bits (rwx); the UID/GID (and effective UID/GID) of the user executing the program stay with it, making system calls to request operating system services (e.g., open/close files, fork another process, create socket for network connections).

Process states and transitions:



Process-related commands:

• ps – report process status (ps shows own processes, “ps aux” shows detailed information of all processes, see below):

```
% ps  aux | more
USER        PID  %CPU  %MEM     SZ   RSS   TT      S   START   TIME    COMMAND
root        597   0.4   0.4   4496  2664   ?       S   Dec 30   31:44   /usr/sbin/nsr/nsr
root          3   0.4   0.0      0     0   ?       S   Dec 30  450:07   fsflush
someone   18420   0.3   0.4   4712  2872   pts/46  S   10:38:26  0:00    pine
root        509   0.2   0.4   4976  2888   ?       S   Dec 30  218:35   /usr/sbin/nsr/nsrd
root        919   0.2   0.8  50984  5344   pts/2   S   Jan 13   55:20 .  /nwadmin
root        253   0.1   0.7   7112  4544   ?       S   Dec 30  413:26   /usr/lib/autofs/au
lang      26671   0.1   0.1   1232   816   pts/3   S   02:46:32 0:02    -csh
root      19144   0.1   0.2   1296  1112   pts/3   O   10:51:20 0:00    ps
auxroot     910   0.0   0.0    288     8   pts/2   T   Jan 13   0:00    sh          daemon
19993      0.0      0.0      0      0          Z            0:00     <defunct>
```

RSS: resident set size (physical memory in Kbytes); TT: the terminal to which the process is attached; S: process status (O: running, S: sleeping, R: runnable, on a ready queue, Z: zombie, T: stopped)

- top – display and update top cpu processes, see example below:

```
% top

last pid: 21646;  load averages:  0.76,  0.36,  0.30                11:30:55
230 processes: 221 sleeping, 7 stopped, 2 on cpu
CPU states: 36.6% idle, 33.7% user,  5.8% kernel, 23.8% iowait,  0.0% swap
Memory: 768M real, 26M free, 946M swap in use, 723M swap free

  PID    USERNAME THR PRI  NICE   SIZE   RES    STATE   TIME    CPU     COMMAND

21606    someone    1    0     0    28M   21M    cpu14     1:01    28.06%   netscape
597      root       1   28    10   4496K 2512K  sleep    32:15     0.70%    nsrindexd
21638    lang       1   58     0   2280K 2096K  cpu10     0:04     0.63%    top
253      root       7   10     0   7112K 4536K   sleep   413:50     0.25%    automountd
598      root       2   59   -15   4232K 2512K   sleep   139:33     0.22%    nsrmmd
509      root       1   58     0   4976K 2784K    sleep   218:49     0.20%    nsrd
```

- kill – terminate a process, see example below:

```
% ps

  PID    TT      S   TIME  COMMAND

21637   pts/3    S    0:00    a.out
26671   pts/3    S    0:02    -csh
26692   pts/3    T   0:08     elm

% kill -9 21637

[2]    Killed              a.out
```

- bg – put the currently stopped process (by CTRL-Z) in the background running
- jobs – show the running or stopped jobs

```
% jobs
[1]  - Stopped (signal)     elm
[2]  + Running              a.out
```

- fg – bring a background job into foreground running

```
% jobs
[1]  - Stopped (signal)     elm
[2]  + Running              a.out
% fg  %2
a.out
```

Note: Information about processes is useful when performing forensic work on "live" systems, e.g., locating processes with strange names, altered system processes, trojan horses.

The /proc file system:

All processes are represented as files located under directory /proc, with each process in a separate subdirectory named by the process id. For example:

```
% ps aux | wc
    227    2672   16655  (that is, there are 227 processes in the system)
% ls /proc | wc
    224     224    1212    (there are 224 files in directory /proc)
% ps
   PID TT       S  TIME COMMAND
25581 pts/3    S  0:00    a.out
26671 pts/3    S  0:03    -csh
26692 pts/3    T  0:08    elm
% ls -ld /proc/25581
dr-x--x--x   5 lang         736 Jan 22 12:36 25581    (the long listing of the directory /proc/25581)
% ls -l  /proc/25581
total 854
-rw-------   1 lang      860160   Jan 22 12:36  as
-r--------   1 lang         152   Jan 22 12:36 auxv
-r--------   1 lang          60   Jan 22 12:36 cred
--w-------   1 lang           0   Jan 22 12:36 ctl
lr-x------   1 lang           0   Jan 22 12:36 cwd ->
dr-x------   2 lang         416   Jan 22 12:36 fd
-r--r--r--   1 lang         120   Jan 22 12:36 lpsinfo
-r--------   1 lang         912   Jan 22 12:36 lstatus
```

(continued next slide)

```
-r--r--r--   1 lang      536   Jan 22 12:36 lusage
dr-xr-xr-x   3 lang       48   Jan 22 12:36 lwp
-r--------   1 lang      960   Jan 22 12:36 map
dr-x------   2 lang      288   Jan 22 12:36 object
-r--------   1 lang     1136   Jan 22 12:36 pagedata
-r--r--r--   1 lang      336   Jan 22 12:36 psinfo
-r--------   1 lang      960   Jan 22 12:36 rma
lr-x------   1 lang        0   Jan 22 12:36 root ->
-r--------   1 lang     1440   Jan 22 12:36 sigact
-r--------   1 lang     1232   Jan 22 12:36 status
-r--r--r--   1 lang      256   Jan 22 12:36 usage
-r--------   1 lang        0   Jan 22 12:36 watch
-r--------   1 lang     1520   Jan 22 12:36 xmap
```

Descriptions of the contents of some of these subdirectories are:

status: process state including PID, size, location

pcinfo: information used by the ps command

as: process virtual address space

fd: a subdirectory containing one entry for each open file

Note: On a Sun OS system (Solaris), there are proc utility tools (under /usr/proc/bin) that extract and return process information presumably held at the /proc directory.

- Mac OS X and Unix articles (tutorial, advanced Unix, how-to's)
- The Law Enforcement and Forensic Examiner's Introduction to Linux, v.3.65 by Barry J. Grundy