Unix Administration

An Advanced Introduction to Unix/C Programming

John Dempsey
COMP-232 Programming Languages
California State University, Channel Islands

What to cover

- **Unix File System, Directories, and Files**
  - **/etc/passwd**
  - **/etc/shadow**
  - **/etc/group**
  - **/etc/hosts**
  - **/etc/hostname**
  - **/etc/services**
  - **/etc/os-release**
  - **/etc/resolv-conf**
  - **/etc/crontab**
  - **/etc/sudoers**
  - **/sbin directory**

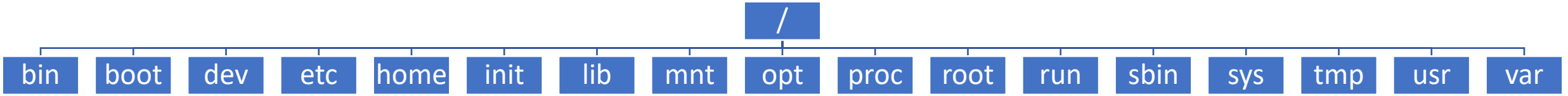
Unix Administrator Commands

- **adduser**
- **su – root**
- **df -k**
- **du –sk ***
- **crontab –l; crontab -e**
- **ipcs**

Add User Manually as root

- **Add user to /etc/passwd file**
- **Add group to /etc/group**
- **mkdir /home/user_name**
- **su – user_name (check login)**

Unix File System



The top node "/" is called the root directory.

john@oho:~$ ls /

```
bin   dev  home  lib    lib64   media  opt   root  sbin  srv  tmp  var
boot  etc  init  lib32  libx32  mnt    proc  run   snap  sys  usr
```

/bin and /sbin Directories

/bin contains most Unix commands that you can run.
Examples are:

ls, more, cat, vi

/sbin contains Unix commands for system administration.

Examples are:

adduser, fsck, mkfs, reboot

To locate where a Unix command resides, run: **which <command>**

/home Directories

john@oho:/home$ cd ← **Change directory into /home/john**

john@oho:~$ ls -a1 ← **List all files in one column**

.

..

.bashrc

.profile

.ssh

.vim

.viminfo

.vimrc

COMP232

a.out

c.c

/tmp Directory

/tmp contains temporary files which are removed when the system is rebooted.

Everyone on the system can write to /tmp.

Makes for a great place to create/view temporary files, like:

```
% crontab -l > /tmp/crontab.txt
% grep MAX_VALUE *.c > /tmp/max_value.txt
```

/usr Directory

/usr contains:

```
john@oho:/usr$ ls -F /usr
bin/    include/  lib32/  libexec/  local/  share/
games/  lib/      lib64/  libx32/   sbin/   src/
```

/usr/include is where the actual include files are located, e.g.

```
#include <stdio.h>
#include <stdlib.h>
```

/usr/local allows you to add local binary, include, lib, and src files:

```
john@oho:/usr/local$ ls -F /usr/local
bin/  etc/  games/  include/  lib/  man@  sbin/  share/  src/
```

/usr/local/bin contains programs local to your system and that others in your group can run.

/etc

/etc contains 186 files and directories. We'll cover some of these.

/etc contains system configuration files, such as:

- hosts
- passwd
- services

and directories, such as:

- X11
- fonts
- init.d

/etc/passwd

- **Defines all users on Unix system.**

Format:

login_name:password:user_id:group_id:user_name:home_directory:shell

where:

Login Name used to log into system.

Password is not used. Passwords are found in /etc/shadow file today.

User Id holds unique numeric value for user.

Group Id holds numeric value for user's primary group.

User Name is a comment field to store first/last name or application name.

Home directory is user's home and location after logging in.

Shell defines the default shell user uses after logging in.

/etc/passwd

john@oho:~$ **more /etc/passwd**

root:x:0:0:root:/root:/bin/bash ← **User id 0/Group id 0 is known as the super user/root user. root user has full access to everything on the system.**

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

bin:x:2:2:bin:/bin:/usr/sbin/nologin

sys:x:3:3:sys:/dev:/usr/sbin/nologin ← **Login ids do not need to be associated with an actual person.**

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/usr/sbin/nologin ← **nologin says you cannot log into the system as user sys.**

man:x:6:12:man:/var/cache/man:/usr/sbin/nologin

lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin

mail:x:8:8:mail:/var/mail:/usr/sbin/nologin ← **Says mail program has /var/mail as its home directory.**

news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin

proxy:x:13:13:proxy:/bin:/usr/sbin/nologin

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

syslog:x:104:110::/home/syslog:/usr/sbin/nologin

sshd:x:109:65534::/run/sshd:/usr/sbin/nologin

john:x:1000:1000:John Dempsey:/home/john:/bin/bash ← **Start of regular user accounts. User id 1000, group id 1000, User John Dempsey, home dir is /home/john, uses /bin/bash**

**** The above list is a partial list of all passwd entries.***

/etc/shadow

- File not readable to most users.

Format:

login_name:password:date_of_last_password_change:min_password_age:maximum_password_age:
password_warning_period:password_inactivity_period:account_expiration_date:reserved_field

Where:

Login name is user login as found in the /etc/passwd file.

Password is the encrypted password for user.

Date of last password change expressed as the number of days since January 1, 1970.

Minimum password age is the number of days the user must wait before being able to change their password.

Maximum password age is the number of days before user must change their password.

Password warning period is the number of days before user is notified they are reaching the maximum age. 0 no warning.

Password inactivity period is the maximum number of days past the maximum age where user can still login and must then change their password. Empty is no expiration period.

Account expiration date is when user can no longer log into system. Empty or 0 means no expiration.

/etc/group

- **Defines all user groups on the Unix system.**
- **When you log into a Unix system, the passwd file sets your default group.**
- **A user is assigned one “primary group”, but may belong to multiple groups.**
- **Each file and directory is owned by one group on the system.**

Fields:

group_name:password:GID:user_list

where:

group_name is the name of the group.
password is the encrypted group password or x if not used.
GID is group id number.
user_list contains list of usernames that are members of this group.

```
john@oho:/etc$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
…
admin:x:116:
netdev:x:117:john
john:x:1000:
staff:x:1001:amy,betty,john,miguel,wendy
```

/etc/hosts

The hosts file contains Internet Protocol (IP) addresses. It's format is:

IP Address hostname aliases …

john@oho:~$ **cat /etc/hosts**

```
127.0.0.1          localhost
127.0.1.1          oho.localdomain oho
192.60.50.10       charlie         prodhost
192.60.50.11       lucy            devhost
192.60.50.15       linus           testhost
143.198.238.179    comp232         comp232.com
142.93.89.28       openhouseon.com
45.55.2.35         plus1se.com     ← If DNS not available, can enter domain names.
```

/etc/services

- To access a service on a system, you need two things:
  1. **The IP address of the system.**
  2. **The port number of the service enabled on the system.**

- /etc/services defines the port number and the transport protocol (TCP and/or UDP) supported for each known service.

- Officially assigned port number are defined by Internet Assigned Numbers Authority (IANA) at https://www.iana.org.

- Companies can add unassigned port numbers to support local applications.

/etc/services

There are 413 lines in the /etc/services file. Here are some of the more important services.

```
% cat /etc/services
tcpmux          1/tcp                           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
netstat         15/tcp
ftp-data        20/tcp
ftp             21/tcp
ssh             22/tcp                          # SSH Remote Login Protocol
telnet          23/tcp
smtp            25/tcp          mail
time            37/tcp          timserver
time            37/udp          timserver
whois           43/tcp          nicname
tftp            69/udp
finger          79/tcp
http            80/tcp          www             # WorldWideWeb HTTP
kerberos        88/tcp          kerberos5 krb5 kerberos-sec     # Kerberos v5
kerberos        88/udp          kerberos5 krb5 kerberos-sec     # Kerberos v5
pop3            110/tcp         pop-3           # POP version 3
ntp             123/udp                         # Network Time Protocol
imap2           143/tcp         imap            # Interim Mail Access P 2 and 4
snmp            161/tcp                         # Simple Net Mgmt Protocol
snmp            161/udp
```

/etc/services Continued

```
snmp-trap       162/tcp         snmptrap        # Traps for SNMP
snmp-trap       162/udp         snmptrap
mailq           174/tcp                         # Mailer transport queue for Zmailer
xdmcp           177/udp                         # X Display Manager Control Protocol
bgp             179/tcp                         # Border Gateway Protocol
smux            199/tcp                         # SNMP Unix Multiplexer
qmtp            209/tcp                         # Quick Mail Transfer Protocol
z3950           210/tcp         wais            # NISO Z39.50 database
ipx             213/udp                         # IPX [RFC1234]
#
# UNIX specific services
#
exec            512/tcp
biff            512/udp         comsat
login           513/tcp
who             513/udp         whod
talk            517/udp
ntalk           518/udp
route           520/udp         router routed   # RIP
rsync           873/tcp
ftps-data       989/tcp                         # FTP over SSL (data)
ftps            990/tcp
telnets         992/tcp                         # Telnet over SSL
imaps           993/tcp                         # IMAP over SSL
pop3s           995/tcp                         # POP-3 over SSL
```

/etc/hostname

/etc/hostname contains the name of the host.

% **cat /etc/hostname**
oho

% **hostname**
oho

% **grep oho /etc/hosts**
127.0.1.1 oho.localdomain oho

/etc/profile

/etc/profile runs each time you and others login and helps set up your environment.

```
john@oho:/etc$ more /etc/profile
if [ "${PS1-}" ]; then
  if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ];
then
    # The file bash.bashrc already sets the default
PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi
```

```
if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
```

/etc/os-release

- os-release provides details on the Unix version you're using.

```
john@oho:/etc$ more os-release
NAME="Ubuntu"
VERSION="20.04.2 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.2 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
```

/etc/timezone

- Displays what time zone system is using.

john@oho:/etc$ **more /etc/timezone**

America/Los_Angeles

/etc/resolv.conf

Lists the IP addresses for Domain Name Servers (DNS) to resolve domain names, like https://plus1se.com

john@oho:/etc$ **cat resolv.conf**

```
nameserver      192.168.1.1          ← IP Version 4 Format
nameserver      2001:1998:f00:1::1   ← IP Version 6 Format
nameserver      2001:1998:f00:2::1
search          lan     example.com
```

crontab

john@oho:~/LAB4/STOCKS$ **crontab -l**

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# m h  dom mon dow   command
```

30 17 1 * * /home/john/LAB4/STOCKS/run_report.bash 2024

To edit crontab using vi, type:

% **export EDITOR=vi**
% **crontab –e**

To view crontab, type:

% **crontab -l**

shmflg = 0644 | IPC_CREAT

To create a shared memory id, we can use:

```
if ((shmid = shmget(key, SHM_SIZE, 0644 | IPC_CREAT)) == -1)
```

But wait, IPC_CREATE is defined as:

```
#define IPC_CREATE  0x200
```

So there is no change?

We want:

```
if ((shmid = shmget(key, SHM_SIZE, 01644)) == -1)
```

Answer

if ((shmid = shmget(key, SHM_SIZE, 0644 | IPC_CREAT)) == -1)

0644 is in octal.
IPC_CREAT is 0x200, but 0x200 is in hex.

0644 = 000 110 100 100
0x200 = 0010 0000 0000

0644	0	0	0	1	1	0	1	0	0	1	0	0
0x200	0	0	1	0	0	0	0	0	0	0	0	0
01644	0	0	1	1	1	0	1	0	0	1	0	0

So there is a change and

01644 = 0644 | IPC_CREAT

Root Directories

/bin	Binaries.	Contains 1,103 binary commands most of which you can run.
/dev	Devices.	Contains device definitions.
/etc	Ecetera.	Contains system configuration files, like password and hosts file.
/home	Home.	Contains user home directories.
/lib	Libraries.	A link to /usr/lib
/mnt	Mount.	Device mount points, e.g. C drive mount point.
/opt	Optional.	Can contain optional files and directories. Third party software.
/proc	Processes.	Contains process information.
/sbin	System Binaries.	Contains system binaries some which you can run.
/tmp	Temp.	Contains temporary files which are removed on system reboot.
/usr	User.	User System Resources (USR) directory.
/var	Variable.	Contains variable length files.

/etc/sudoers

- **Lists users who can use sudo to run commands a super user (su).**

john@oho:/etc$ **sudo cat sudoers**
[sudo] password for john:
Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

User privilege specification
root ALL=(ALL:ALL) ALL

Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

#includedir /etc/sudoers.d

john@oho:/etc$ **grep admin /etc/group**
admin:x:116:
john@oho:/etc$ **grep sudo /etc/group**
sudo:x:27:john

adduser

```
root@oho:~# adduser linda
Adding user `linda' ...
Adding new group `linda' (1001) ...
Adding new user `linda' (1003) with group `linda' ...
Creating home directory `/home/linda' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for linda
Enter the new value, or press ENTER for the default
    Full Name []: Linda Day
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
root@oho:~# grep linda /etc/passwd
linda:x:1003:1001:Linda Day,,,:/home/linda:/bin/bash
root@oho:~# grep 1001 /etc/group
linda:x:1001:
```