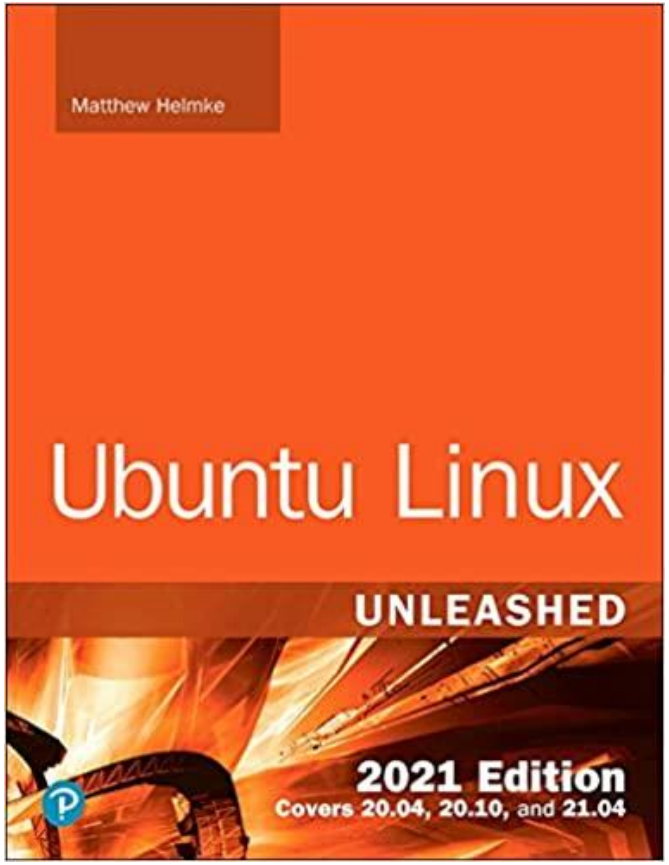
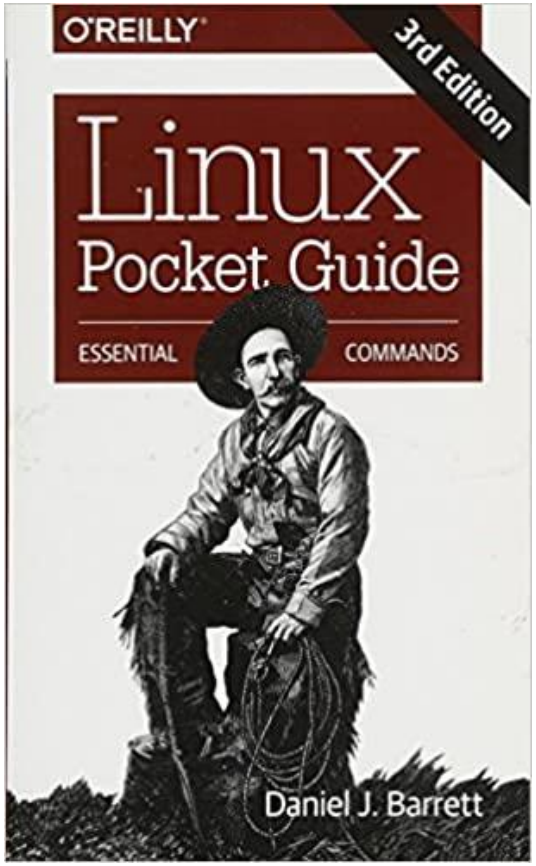
LINUX AND OPEN SOURCE SOFTWARE

CH3. COMMAND LINE

Syllabus & Text-books

1) Matthew Helmke, *Ubuntu Linux unleashed*, Pearson, 2021 Edition.
2) Daniel J. Barrett, *Linux pocket guide*, 3rd edition, O'Reilly, June 2016.





Main contents

1. What is the command line?
2. Accessing the Command Line
3. User Accounts
4. Reading Documentation
5. **Understanding the Linux File System hierarchy**
6. **Navigating the Linux File System**
7. **Working with Permissions**
8. Working with Files
9. Working as Root
10. Commonly Used commands and programs
11. Using basic commands
12. Using advanced commands

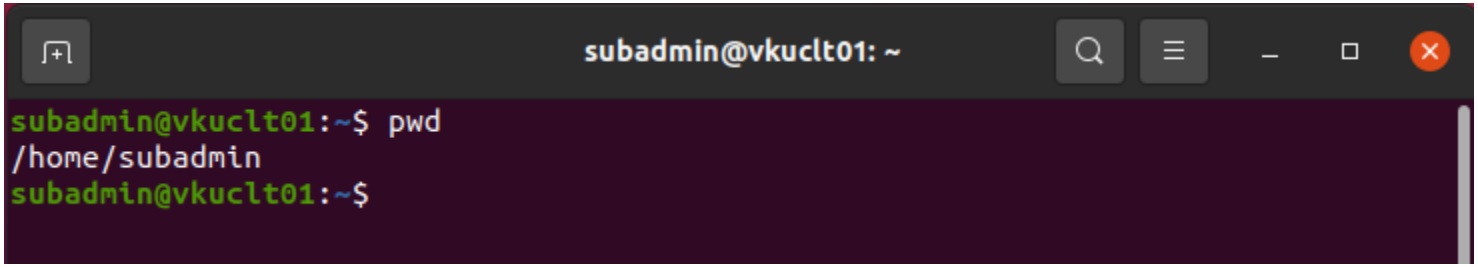
Gửi mail kết quả thực hiện đến ttson@vku.udn.vn

What is the command line?

- GUI stands for Graphical User Interface.
- CLI stands for Command Line Interface.
  - Sometimes things go wrong, and you might not have the luxury of a graphical interface to work with. In such situations, a fundamental understanding of the command line and its uses can be a real lifesaver.
  - Some basic commands:
    - *Routine tasks:* Log in, log out, changing passwords, etc.
    - *Basic file management:* Create file and folder, copy, paste, rename, etc.
    - *Basic system management:* shutting down, rebooting, file permissions, etc.

Accessing the Command Line

- From GUI, run *gnome-terminal* by press Ctrl + Alt + T



- *subadmin* is the user who is logged in the system.
- *The tilde (~)* as shorthand for the home directory, which would usually be something like */home/subadmin*.
- The command *pwd (print working directory)* is used to get the full path of your location.

Accessing the Command Line

- Logging out
  - Use the *exit* or *logout* command or press Ctrl + D to exit your session.
- Logging In and Out from a Remote Computer
  - You can log in to your computer via a network connection from a remote computer.
  - The best and most secure way to log in to a remote Linux computer is to use *ssh*.

```
matthew@seymour:~$ ssh 192.168.0.41
The authenticity of host '192.168.0.41 (192.168.0.41)' can't be established.
RSA key fingerprint is e1:db:6c:da:3f:fc:56:1b:52:f9:94:e0:d1:1d:31:50.
Are you sure you want to continue connecting (yes/no)?

yes

Warning: Permanently added '192.168.0.41' (RSA) \
to the list of known hosts.
matthew@192.168.0.41's password:
matthew@babbage-$
```

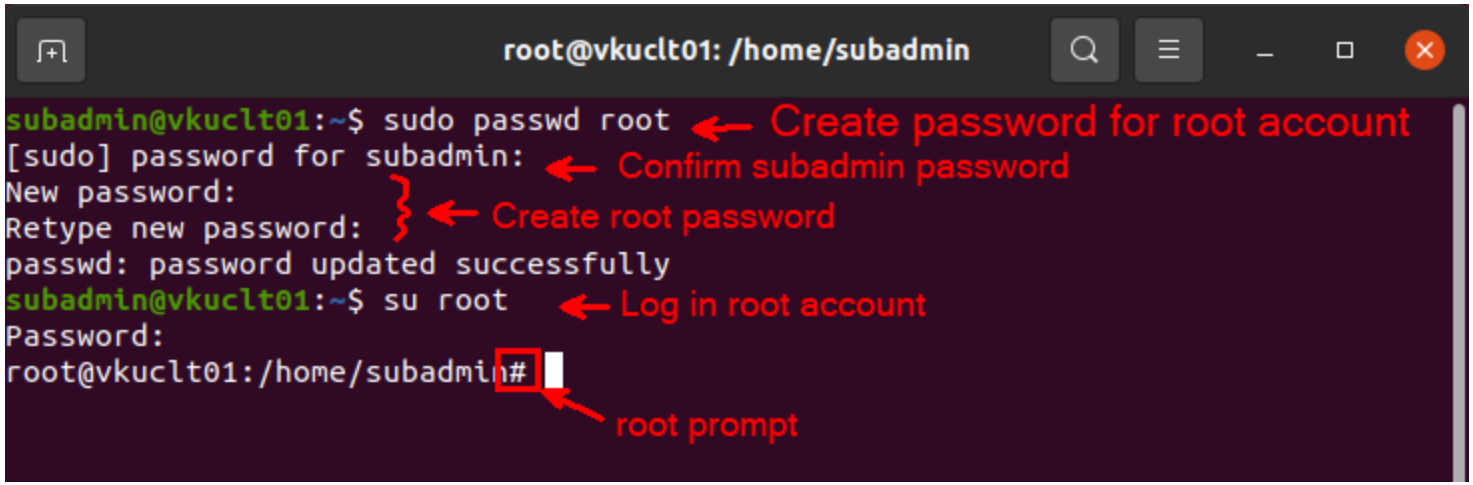
```
matthew@babbage-$ logout
matthew@seymour:~$
```

User Accounts

- User Account = system users + human users
- Human users = regular users + super user privileges
  - *Regular user* can change anything that is specific to their accounts, such as the wallpaper on the desktop, their personal preferences, and the configuration for a program when it is run by them using their account.
  - *Super user privileges* can access to the entire system and can carry out any task - even destructive tasks.
    - To use super user privileges, you use *sudo* followed by a space and the command you want to run.

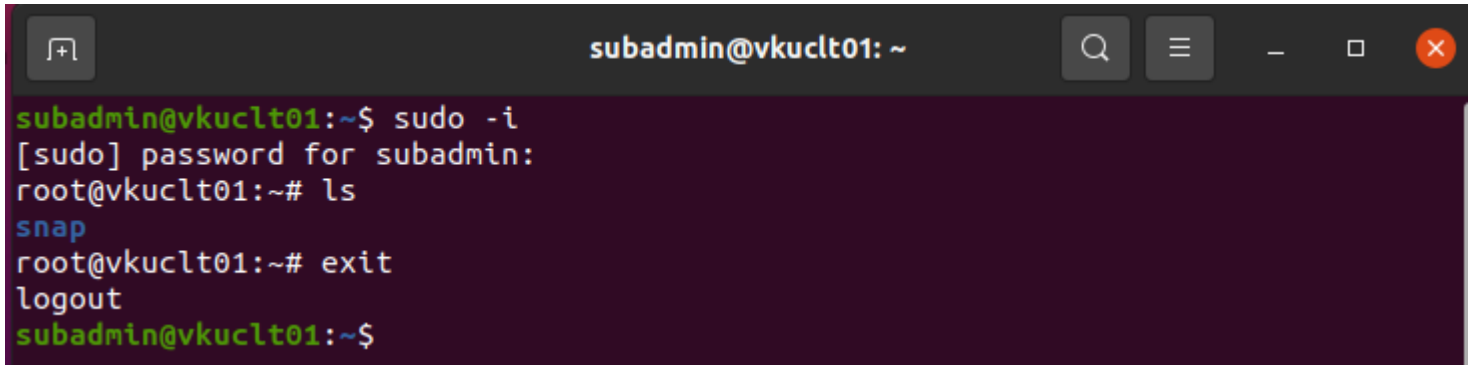
User Accounts

- *Super user privileges*
  - *sudo rm -rf /*, which would erase everything on your hard drive if it did not require appending *--no-preserve-root* to work. Do not try this particular command as a test.
  - *root* is a super user account.
  - In Ubuntu the root account is disabled by default because forcing regular users with super user privileges to type a specific command every time they want to execute a command as a super user.



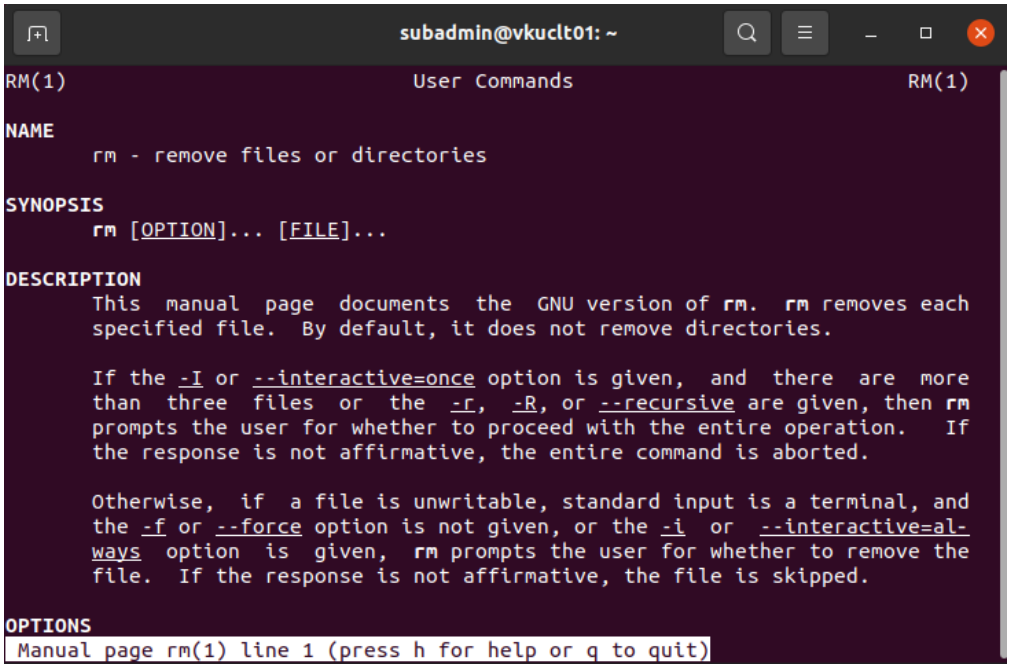
User Accounts

- *Super user privileges*
  - An alternative way of getting a root prompt, without having to enable the root account, is to issue the command *sudo -i*.
  - After entering your password, you find yourself at a root prompt (#).
  - Do what you need to do, and when you are finished, type *exit* and press Enter to return to your usual prompt.



Reading Documentation

- Using Man pages
  - To learn more about a command or program, use the *man* command followed by the name of the command.
  - Man pages are stored in places like */usr/share/man* and */usr/local/share/man*
    - *$man rm*



Reading Documentation

- Using Man pages
  - Another way to get help is used *info* command.
    - *$info rm*

```
subadmin@vkuclt01: ~
Next: shred invocation,  Prev: mv invocation,  Up: Basic operations

11.5 ‘rm’: Remove files or directories
======================================

‘rm’ removes each given FILE.  By default, it does not remove
directories.  Synopsis:

     rm [OPTION]... [FILE]...

   If the ‘-I’ or ‘--interactive=once’ option is given, and there are
more than three files or the ‘-r’, ‘-R’, or ‘--recursive’ are given,
then ‘rm’ prompts the user for whether to proceed with the entire
operation.  If the response is not affirmative, the entire command is
aborted.

   Otherwise, if a file is unwritable, standard input is a terminal, and
the ‘-f’ or ‘--force’ option is not given, or the ‘-i’ or
‘--interactive=always’ option _is_ given, ‘rm’ prompts the user for
whether to remove the file.  If the response is not affirmative, the
file is skipped.

-----Info: (coreutils)rm invocation, 118 lines --Top----------------------------
Welcome to Info version 6.7.  Type H for help, h for tutorial.
```

Reading Documentation

- Using *apropos*
  - You can use the *apropos* to get help
    - *$apropos partition* is used to find commands related to partitioning.

```
subadmin@vkuclt01: ~
subadmin@vkuclt01:~$ apropos partition
addpart (8)          - tell the kernel about the existence of a partition
cfdisk (8)           - display or manipulate a disk partition table
cgdisk (8)           - Curses-based GUID partition table (GPT) manipulator
delpart (8)          - tell the kernel to forget about a partition
fdisk (8)            - manipulate disk partition table
fixparts (8)         - MBR partition table repair utility
gdisk (8)            - Interactive GUID partition table (GPT) manipulator
mpartition (1)       - partition an MSDOS hard disk
parted (8)           - a partition manipulation program
partprobe (8)        - inform the OS of partition table changes
partx (8)            - tell the kernel about the presence and numbering of on...
resizepart (8)       - tell the kernel about the new size of a partition
sfdisk (8)           - display or manipulate a disk partition table
sgdisk (8)           - Command-line GUID partition table (GPT) manipulator fo...
systemd-gpt-auto-generator (8) - Generator for automatically discovering and ...
subadmin@vkuclt01:~$
```

Reading Documentation

- Using *whereis*
  - To find a command and its documentation, you can use the *whereis* command.
  - For example:

```
subadmin@vkuclt01: ~
subadmin@vkuclt01:~$ whereis fdisk
fdisk: /usr/sbin/fdisk /usr/share/man/man8/fdisk.8.gz
subadmin@vkuclt01:~$ whereis rm
rm: /usr/bin/rm /usr/share/man/man1/rm.1.gz
subadmin@vkuclt01:~$
```

Excercise

1. Read document for *useradd* command
2. Create a new user name as your name (or ID) using *useradd* with a password for log-in
3. Log-out and log-in with new created user and check log file of system, auth log in */var/log*
4. Check home directory of the log-on user
5. Change password of the user
6. Change password of the root user

Understanding the Linux File System hierarchy

- Linux has inherited from UNIX a well-planned hierarchy for organizing things.

Table 10.1 Basic Linux Directories

Directory	Description
/	The root directory
/bin	Essential commands
/boot	Boot loader files, Linux kernel
/dev	Device files
/etc	System configuration files
/home	User home directories
/lib	Shared libraries, kernel modules
/lost+found	Recovered files (if found after a file system check)
/media	Mount point for removable media, such as DVDs and floppy disks
/mnt	Usual mount point for local, remote file systems, file systems that are additional to the standard, such as a DVD-ROM or another HDD
/opt	Add-on software packages
/proc	Kernel information, process control
/root	Super user (root) home
/sbin	System commands (mostly root only)
/sys	Real-time information on devices used by the kernel
/tmp	Temporary files
/usr	Software not essential for system operation, such as applications
/var	Variable files relating to services that run on the system, but whose contents are expected to change regularly during normal operation

Understanding the Linux File System hierarchy

- Using *$man hier* to show Linux File system hierarchy

```
subadmin@vkuclt01: ~
HIER(7)                    Linux Programmer's Manual                    HIER(7)

NAME
       hier - description of the filesystem hierarchy

DESCRIPTION
       A typical Linux system has, among others, the following directories:

       /      This  is  the  root  directory.   This  is  where the whole tree
              starts.

       /bin   This directory contains executable programs which are needed  in
              single user mode and to bring the system up or repair it.

       /boot  Contains static files for the boot loader.  This directory holds
              only the files which are needed during the  boot  process.   The
              map  installer  and  configuration  files should go to /sbin and
              /etc.  The operating system kernel (initrd for example) must  be
              located in either / or /boot.

       /dev   Special  or  device files, which refer to physical devices.  See
              mknod(1).

 Manual page hier(7) line 1 (press h for help or q to quit)
```

Note: list the current file system: Using *tree* command

Understanding the Linux File System hierarchy

- **Essential commands** in */bin* and */sbin*
  - The */bin* directory contains essential commands used by the system for running and booting the system.
  - The */sbin* directory is used by root operators
- **Configuration Files in */etc***
  - *fstab* - the file system table is a text file that lists each hard drive, CD-ROM, or other storage device attached to your PC.
  - *modprobe.d/* - This folder holds all the instructions to load kernel modules that are required as part of system startup.
  - *passwd* - This file holds the list of users for the system.
  - *sudoers* - This file holds a list of users or user groups with super user access.

Understanding the Linux File System hierarchy

- **User Directories: */home***
  - User directories are named by default according to account usernames.
  - For example, if you have an account named *matthew*, your home directory would generally be found in */home/matthew*.
- **Using the Contents of the *proc*** directory to interact with or obtain information from the Kernel
  - The contents of */proc* directory are created from memory and exit only while Linux is running.
  - For example, the *free* command obtains its information from a file named *meminfo*:

```
subadmin@vkuclt01: ~
subadmin@vkuclt01:~$ free
              total        used        free      shared  buff/cache   available
Mem:        3984952     1888620      413348       76428     1682984     1735040
Swap:       1918356      298232     1620124
subadmin@vkuclt01:~$
```

Understanding the Linux File System hierarchy

- Using the Contents of the */proc* directory to interact with or obtain information from the Kernel
  - To see the contents of the *meminfo* file:

```
subadmin@vkuclt01: ~
subadmin@vkuclt01:~$ cat /proc/meminfo
MemTotal:        3984952 kB
MemFree:          410324 kB
MemAvailable:    1733716 kB
Buffers:           74952 kB
Cached:          1427052 kB
SwapCached:        17800 kB
Active:          1009688 kB
Inactive:        1792196 kB
Active(anon):     223664 kB
Inactive(anon):  1148460 kB
Active(file):     786024 kB
Inactive(file):   643736 kB
Unevictable:          32 kB
Mlocked:              32 kB
SwapTotal:       1918356 kB
SwapFree:        1621404 kB
Dirty:                28 kB
Writeback:             0 kB
```

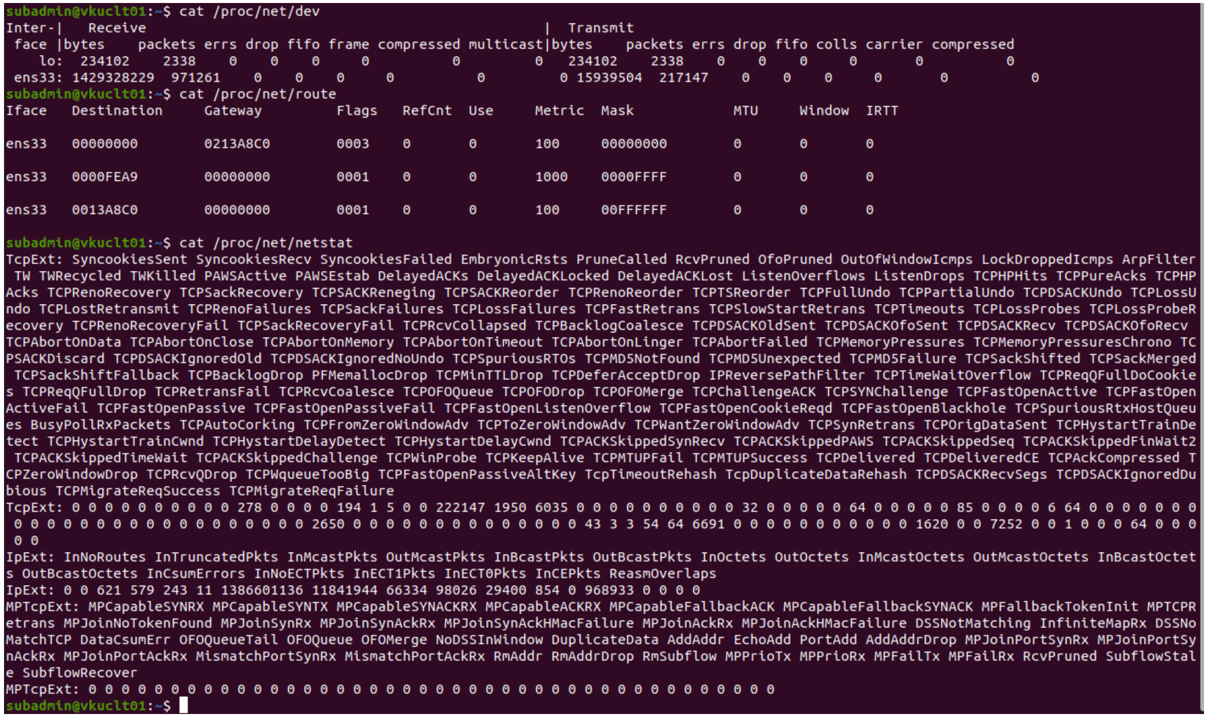
Understanding the Linux File System hierarchy

- Using the Contents of the */proc* directory to interact with or obtain information from the Kernel
  - Getting CPU information, such as the family, type, and speed from */proc/cpuinfo*

```
subadmin@vkuclt01: ~
subadmin@vkuclt01:~$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 158
model name      : Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz
stepping        : 9
microcode       : 0xb4
cpu MHz         : 3000.005
cache size      : 6144 KB
```

Understanding the Linux File System hierarchy

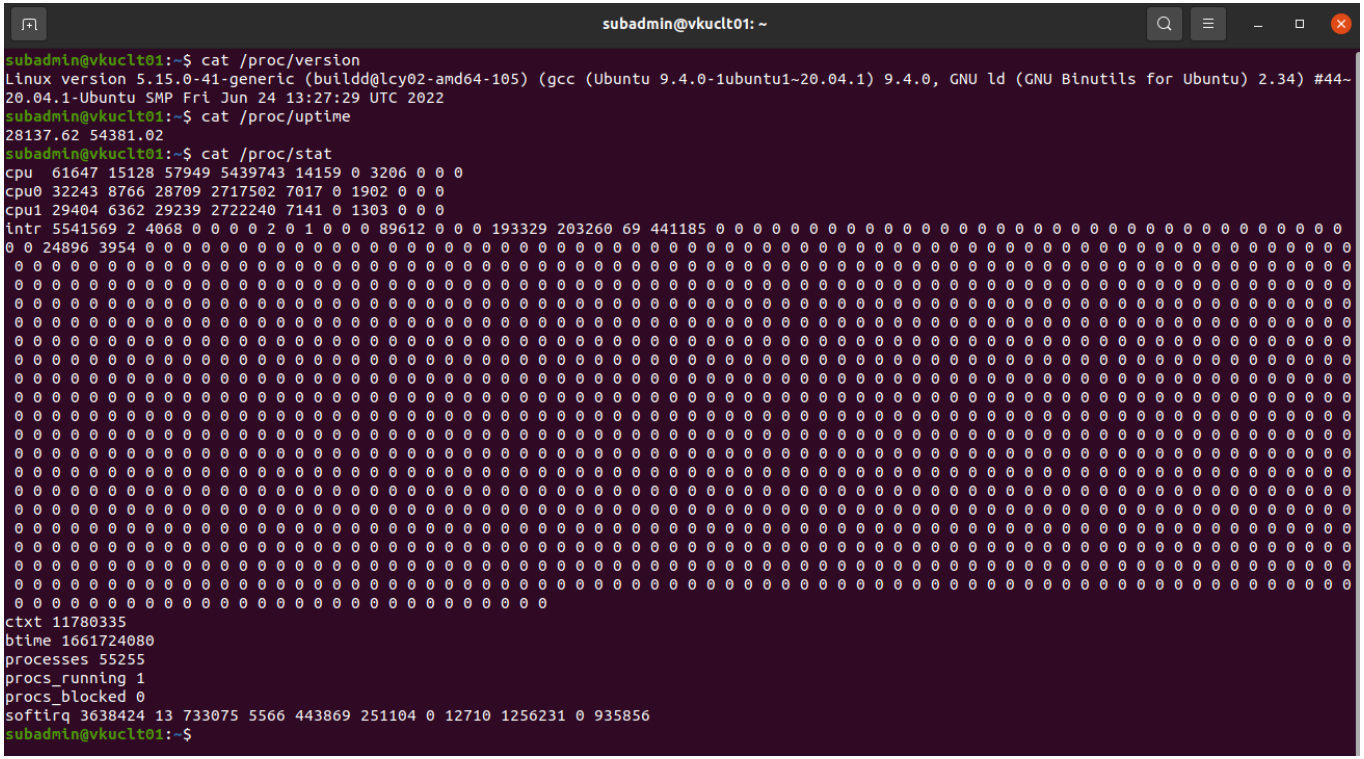
- Using the Contents of the */proc* directory to interact with or obtain information from the Kernel
  - Viewing important networking information under */proc/net*, */proc/net/dev*, routing information in */proc/net/route* and network statistics in */proc/net/netstat*



```
subadmin@vkuclt01:~$ cat /proc/net/dev
Inter-|   Receive                                                |  Transmit
 face |bytes    packets errs drop fifo frame compressed multicast|bytes    packets errs drop fifo colls carrier compressed
    lo:  234102    2338    0    0    0     0          0         0   234102    2338    0    0    0     0       0          0
 ens33: 1429328229  971261    0    0    0     0          0         0 15939504  217147    0    0    0     0       0          0
subadmin@vkuclt01:~$ cat /proc/net/route
Iface   Destination     Gateway         Flags   RefCnt  Use     Metric  Mask            MTU     Window  IRTT

ens33   00000000        0213A8C0        0003    0       0       100     00000000        0       0       0

ens33   0000FEA9        00000000        0001    0       0       1000    0000FFFF        0       0       0

ens33   0013A8C0        00000000        0001    0       0       100     00FFFFFF        0       0       0

subadmin@vkuclt01:~$ cat /proc/net/netstat
TcpExt: SyncookiesSent SyncookiesRecv SyncookiesFailed EmbryonicRsts PruneCalled RcvPruned OfoPruned OutOfWindowIcmps LockDroppedIcmps ArpFilter
 TW TWRecycled TWKilled PAWSActive PAWSEstab DelayedACKs DelayedACKLocked DelayedACKLost ListenOverflows ListenDrops TCPHPHits TCPPureAcks TCPHP
Acks TCPRenoRecovery TCPSackRecovery TCPSACKReneging TCPSACKReorder TCPRenoReorder TCPTSReorder TCPFullUndo TCPPartialUndo TCPDSACKUndo TCPLossU
ndo TCPLostRetransmit TCPRenoFailures TCPSackFailures TCPLossFailures TCPFastRetrans TCPSlowStartRetrans TCPTimeouts TCPLossProbes TCPLossProbeR
ecovery TCPRenoRecoveryFail TCPSackRecoveryFail TCPRcvCollapsed TCPBacklogCoalesce TCPDSACKOldSent TCPDSACKOfoSent TCPDSACKRecv TCPDSACKOfoRecv
TCPAbortOnData TCPAbortOnClose TCPAbortOnMemory TCPAbortOnTimeout TCPAbortOnLinger TCPAbortFailed TCPMemoryPressures TCPMemoryPressuresChrono TC
PSACKDiscard TCPDSACKIgnoredOld TCPDSACKIgnoredNoUndo TCPSpuriousRTOs TCPMD5NotFound TCPMD5Unexpected TCPMD5Failure TCPSackShifted TCPSackMerged
 TCPSackShiftFallback TCPBacklogDrop PFMemallocDrop TCPMinTTLDrop TCPDeferAcceptDrop IPReversePathFilter TCPTimeWaitOverflow TCPReqQFullDoCookie
s TCPReqQFullDrop TCPRetransFail TCPRcvCoalesce TCPOFOQueue TCPOFODrop TCPOFOMerge TCPChallengeACK TCPSYNChallenge TCPFastOpenActive TCPFastOpen
ActiveFail TCPFastOpenPassive TCPFastOpenPassiveFail TCPFastOpenListenOverflow TCPFastOpenCookieReqd TCPFastOpenBlackhole TCPSpuriousRtxHostQueu
es BusyPollRxPackets TCPAutoCorking TCPFromZeroWindowAdv TCPToZeroWindowAdv TCPWantZeroWindowAdv TCPSynRetrans TCPOrigDataSent TCPHystartTrainDe
tect TCPHystartTrainCwnd TCPHystartDelayDetect TCPHystartDelayCwnd TCPACKSkippedSynRecv TCPACKSkippedPAWS TCPACKSkippedSeq TCPACKSkippedFinWait2
 TCPACKSkippedTimeWait TCPACKSkippedChallenge TCPWinProbe TCPKeepAlive TCPMTUPFail TCPMTUPSuccess TCPDelivered TCPDeliveredCE TCPAckCompressed T
CPZeroWindowDrop TCPRcvQDrop TCPWqueueTooBig TCPFastOpenPassiveAltKey TcpTimeoutRehash TcpDuplicateDataRehash TCPDSACKRecvSegs TCPDSACKIgnoredDu
bious TCPMigrateReqSuccess TCPMigrateReqFailure
TcpExt: 0 0 0 0 0 0 0 0 0 0 278 0 0 0 0 194 1 5 0 0 222147 1950 6035 0 0 0 0 0 0 0 0 0 0 32 0 0 0 0 0 64 0 0 0 0 0 85 0 0 0 0 6 64 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2650 0 0 0 0 0 0 0 0 0 0 0 0 0 0 43 3 3 54 64 6691 0 0 0 0 0 0 0 0 0 0 1620 0 0 7252 0 0 1 0 0 0 64 0 0 0
 0 0
IpExt: InNoRoutes InTruncatedPkts InMcastPkts OutMcastPkts InBcastPkts OutBcastPkts InOctets OutOctets InMcastOctets OutMcastOctets InBcastOctet
s OutBcastOctets InCsumErrors InNoECTPkts InECT1Pkts InECT0Pkts InCEPkts ReasmOverlaps
IpExt: 0 0 621 579 243 11 1386601136 11841944 66334 98026 29400 854 0 968933 0 0 0 0
MPTcpExt: MPCapableSYNRX MPCapableSYNTX MPCapableSYNACKRX MPCapableACKRX MPCapableFallbackACK MPCapableFallbackSYNACK MPFallbackTokenInit MPTCPR
etrans MPJoinNoTokenFound MPJoinSynRx MPJoinSynAckRx MPJoinSynAckHMacFailure MPJoinAckRx MPJoinAckHMacFailure DSSNotMatching InfiniteMapRx DSSNo
MatchTCP DataCsumErr OFOQueueTail OFOQueue OFOMerge NoDSSInWindow DuplicateData AddAddr EchoAdd PortAdd AddAddrDrop MPJoinPortSynRx MPJoinPortSy
nAckRx MPJoinPortAckRx MismatchPortSynRx MismatchPortAckRx RmAddr RmAddrDrop RmSubflow MPPrioTx MPPrioRx MPFailTx MPFailRx RcvPruned SubflowStal
e SubflowRecover
MPTcpExt: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
subadmin@vkuclt01:~$
```

Understanding the Linux File System hierarchy

- Using the Contents of the *\/proc* directory to interact with or obtain information from the Kernel
  - Getting the kernel version in *\/proc/version*
  - Performance information uptime in *\/proc/uptime*
  - Show statistics such as CPU load, swap file usage, and processes in *\/proc/stat*.



```
subadmin@vkuclt01:~$ cat /proc/version
Linux version 5.15.0-41-generic (buildd@lcy02-amd64-105) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #44~
20.04.1-Ubuntu SMP Fri Jun 24 13:27:29 UTC 2022
subadmin@vkuclt01:~$ cat /proc/uptime
28137.62 54381.02
subadmin@vkuclt01:~$ cat /proc/stat
cpu  61647 15128 57949 5439743 14159 0 3206 0 0 0
cpu0 32243 8766 28709 2717502 7017 0 1902 0 0 0
cpu1 29404 6362 29239 2722240 7141 0 1303 0 0 0
intr 5541569 2 4068 0 0 0 0 2 0 1 0 0 0 89612 0 0 0 193329 203260 69 441185 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 24896 3954 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ctxt 11780335
btime 1661724080
processes 55255
procs_running 1
procs_blocked 0
softirq 3638424 13 733075 5566 443869 251104 0 12710 1256231 0 935856
subadmin@vkuclt01:~$
```

Understanding the Linux File System hierarchy

- Working with Shared Data in the */usr* directory
  - The */usr* directory contains software applications, libraries, and other types of shared data for use by anyone on the system.
    - */usr/share/man*: manual pages.
    - */usr/share/name_of_package*: Software package shared files.
    - */usr/share/doc*: Software package documentation.
    - */usr/local:* Locally built and installed software

Understanding the Linux File System hierarchy

- Temporary File Storage in the */tmp* directory
  - The */tmp* directory is used for temporary file storage.
  - Files in this directory are cleared daily by a *cron* job and every time the system is booted.

```
subadmin@vkuclt01: ~
subadmin@vkuclt01:~$ ls /tmp/
config-err-a4CGAJ                                                    tmpaddon
snap.snap-store                                                      tmpbwzdu316
ssh-XgXQIC034igx                                                     tracker-extract-files.1000
systemd-private-1fcd215b3a264e85a6c4e95d47fb83f2-colord.service-lhMiJi   tracker-extract-files.125
systemd-private-1fcd215b3a264e85a6c4e95d47fb83f2-ModemManager.service-qdcO6g   VMwareDnD
systemd-private-1fcd215b3a264e85a6c4e95d47fb83f2-switcheroo-control.service-l4AHCf   vmware-root_53276-1796424125
systemd-private-1fcd215b3a264e85a6c4e95d47fb83f2-systemd-logind.service-TNT5dh   vmware-root_53319-879750047
systemd-private-1fcd215b3a264e85a6c4e95d47fb83f2-systemd-resolved.service-2kvyci   vmware-root_731-4248811549
systemd-private-1fcd215b3a264e85a6c4e95d47fb83f2-upower.service-sfkgkh
subadmin@vkuclt01:~$
```

Understanding the Linux File System hierarchy

- Accessing Variable Data Files in the /**var** Directory
  - The /var directory contains subdirectories used by various system services for spooling and logging.

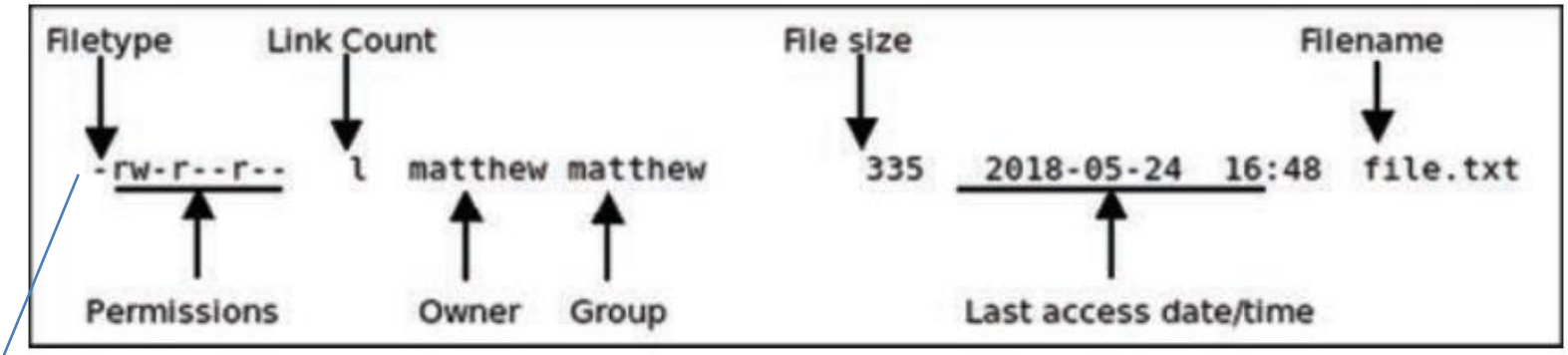
```
ubuntu@ubuntu-virtual-machine:~$ tail -f /var/log/syslog
Sep 14 19:08:46 ubuntu-virtual-machine whoopsie[945]: [19:08:46] online
Sep 14 19:08:53 ubuntu-virtual-machine systemd[1457]: tracker-extract.service: Succeeded.
Sep 14 19:08:57 ubuntu-virtual-machine systemd[1]: NetworkManager-dispatcher.service: Succeeded.
Sep 14 19:09:13 ubuntu-virtual-machine tracker-store[3789]: OK
Sep 14 19:09:13 ubuntu-virtual-machine systemd[1457]: tracker-store.service: Succeeded.
Sep 14 19:11:19 ubuntu-virtual-machine cron[4092]: (CRON) DEATH (can't open or create /var/run/crond.pid: Permission deni
ed)
Sep 14 19:11:23 ubuntu-virtual-machine cron[4095]: (CRON) DEATH (can't lock /var/run/crond.pid, otherpid may be 759: Reso
urce temporarily unavailable)
Sep 14 19:13:46 ubuntu-virtual-machine PackageKit: daemon quit
Sep 14 19:13:46 ubuntu-virtual-machine systemd[1]: packagekit.service: Succeeded.
Sep 14 19:17:01 ubuntu-virtual-machine CRON[4120]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
```

Exercise

- List all folder in the Ubuntu and provide comparison with the Windows file system
  - Tip: using *tree* command
- Check the file system of Ubuntu in *etc/fstab* file
  - Explain each file system
- Check the information of Version, CPU, Mem, Uptime in */proc*
- Check the transmitted, received packets in the */proc/net/dev*

Navigating the Linux File System

- **List the contents of a directory:** *ls - l*



"-" a plain file,
d a directory,
c a character device (such as /dev/ttyS0),
l a symbolic link,
b is used for a block device (such as /dev/sda).

Owner	Group	Others
rwx	rwx	rxw

- **See more detail in page 117, chapter 10**

Working with Permissions

- **$chmod command to alter file permissions**
  - $ *chmod u+rw readme.txt*
  - $ *chmod 666 readme.txt*

  (666 is default for file and 777 is defaults for folder)

  4 read, 2 write, 1 execute

- `u`—Adds or removes user (owner) read, write, or execute permission
- `g`—Adds or removes group read, write, or execute permission
- `o`—Adds or removes read, write, or execute permission for others not in a file's group
- `a`—Adds or removes read, write, or execute permission for all users
- `r`—Adds or removes read permission
- `w`—Adds or removes write permission
- `x`—Adds or removes execution permission

- **See more detail in page 120, chapter 10**

Working with Files

- *touch, rm, mv, cp, cat, less, more*
- *mkdir, rmdir*

- **See more detail in page 128, chapter 10**

Working as Root

- *sudo*
- *#*

- **See more detail in page 133, chapter 10**

Using basic commands

Why Use the Command Line?

Moving from the GUI to the command line is a conscious choice for most people, although it is increasingly rare that it is an absolute choice accompanied by complete abandonment of GUIs.

Reasons for using the command line include the following:

- You want to chain together two or more commands.
- You want to use a command or parameter available only on the shell.
- You are working on a text-only system.
- You have used it for a long time and feel comfortable there.
- You want to automate a task.

Using basic commands

- `cat`—Prints the contents of a file
- `cd`—Changes directories
- `chmod`—Changes file access permissions
- `cp`—Copies files
- `du`—Prints disk usage
- `emacs`—Edits text files
- `find`—Finds files by searching
- `grep`—Searches for a string in input or files
- `head`—Prints the first lines of a file
- `less`—Displays files or input interactively
- `ln`—Creates links between files
- `locate`—Finds files from an index
- `ls`—Lists files in the current directory

Using basic commands

- `make`—Compiles and installs programs
- `man`—Displays manual pages for reading
- `mkdir`—Makes directories
- `mv`—Moves files
- `nano`—Edits text files
- `rm`—Deletes files and directories
- `sort`—Takes a text file as input and outputs the contents of the file in the order you specify
- `ssh`—Connects to other machines using a secure shell connection
- `tail`—Prints the last lines of a file
- `vim`—Edits text files
- `which`—Prints the location of a command

➢ **See more detail in page 143, chapter 11**

Using advanced commands

- $ *diff file1 file2:* differences between two files
- $ *comm file1 file2*: common between two files
- $*cat /proc/cpuinfo > file.txt*
- $*cat < file.txt*

Stream	Abbreviation	Number
Standard input	stdin	0
Standard output	stdout	1
Standard error, or error stream	stderr	2

Using advanced commands

- It also looks at the three most popular **Linux text editors**: *vim, emacs,* and *nano*, as well as the *sed* and *awk* tools.
  - **emacs**—The comprehensive GNU emacs editing environment, which is much more than an editor; see the section "Working with emacs," later in this chapter
  - **nano**—A simple text editor similar to the classic pico text editor that was included with the once-common pine email program
  - **vim**—An improved compatible version of the vi text editor (which we call vi in the rest of this chapter because it has a symbolic link named vi and a symbolically linked manual page)
- Graphical text editors
  - **gedit**—A GUI text editor for GNOME, which is installed by default with Ubuntu
  - **kate**—A simple KDE text editor
  - **kedit**—Another simple KDE text editor
- **See more detail in page 185, chapter 12**

Redirecting Output and Input

matthew@seymour:~$ **cat /proc/cpuinfo > file.txt**

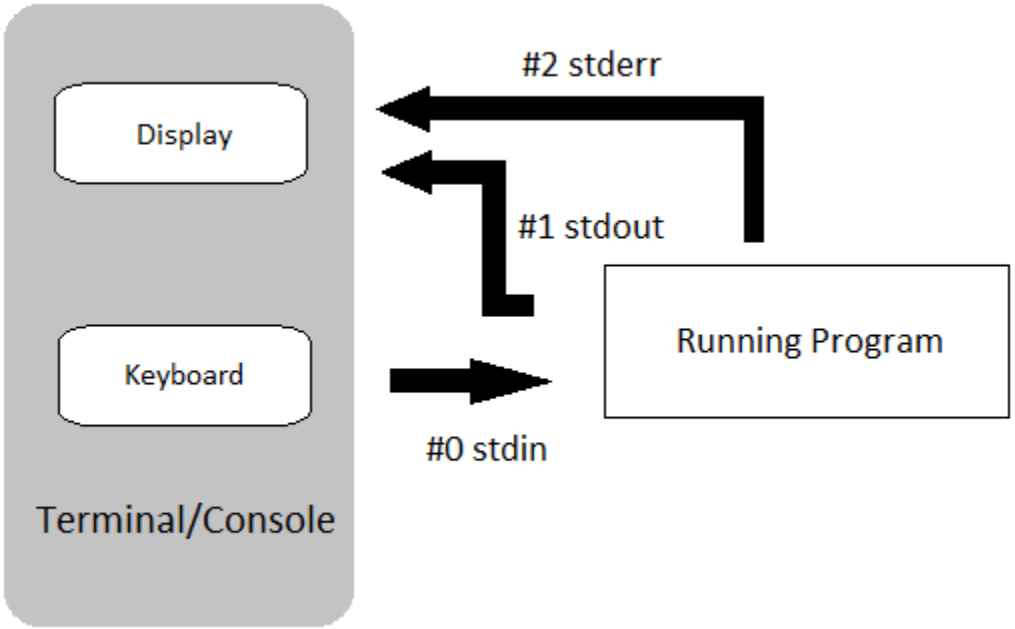
matthew@seymour:~$ **cat < file.txt**

matthew@seymour:~$ **sudo dpkg --get-selections > pkg.list**

matthew@seymour:~$ **sudo dpkg --set-selections < pkg.list**

matthew@seymour:~$ **echo "This is a new line being added." >> file.txt**

stdin, stdout, stderr, and Redirection



Stream	Abbreviation	Number
Standard input	stdin	0
Standard output	stdout	1
Standard error or Error stream	stderr	2

stdin, stdout, stderr, and Redirection

matthew@seymour:~$ ***program* 2> error.log**

matthew@seymour:~$ ***program*** &> *filename*
matthew@seymour:~$ ***program* >> *filename* 2>&1**

matthew@seymour:~$ *program* **2>&1**

Comparing Files

- Finding Differences in Files with **diff**

matthew@seymour:~$ **diff** ***file1 file2***

- Finding Similarities in Files with **comm**

matthew@seymour:~$ **comm** ***file1 file2***

Limiting Resource Use and Job Control

- Listing Processes with **ps**

```
matthew@seymour:~$ ps -aux –sort=%cpu
```

sort by: command, %cpu, pid, and user

- Listing Jobs with **jobs**

```
matthew@seymour:~$ jobs
```

- Running One or More Tasks in the Background

```
matthew@seymour:~$ command &
[1] 11423
```

Limiting Resource Use and Job Control

- Printing Resource Usage with **top**

```
top - 18:10:40 up 48 min,  2 users,  load average: 0.73, 0.83, 0.49
Tasks: 135 total,   1 running, 134 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.3%us,  0.3%sy,  0.0%ni, 99.0%id,  0.0%wa,  0.0%hi,  0.3%si,  0.0%st
Mem:    508488k total,   393612k used,   114876k free,    54008k buffers
Swap:   916476k total,        0k used,   916476k free,   180944k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1027 root      20   0  163m  21m 7960 S  0.3  4.4   0:18.07 Xorg
 1371 root      20   0  5808 2868 2332 S  0.3  0.6   0:05.74 vmtoolsd
 1764 matthew   20   0 46564  20m  16m S  0.3  4.1   0:13.12 vmware-user-loa
 2916 matthew   20   0 61572  13m  10m S  0.3  2.7   0:00.45 gnome-terminal
 2941 matthew   20   0  2624 1116  840 R  0.3  0.2   0:00.20 top
    1 root      20   0  2868 1700 1224 S  0.0  0.3   0:01.45 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S  0.0  0.0   0:00.16 ksoftirqd/0
    4 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    6 root      20   0     0    0    0 S  0.0  0.0   0:00.12 events/0
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.00 cpuset
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.00 khelper
    9 root      20   0     0    0    0 S  0.0  0.0   0:00.00 netns
   10 root      20   0     0    0    0 S  0.0  0.0   0:00.00 async/mgr
```

- Setting Process Priority with **nice**

```
matthew@seymour:~$ sudo nice -n 19 tar czf compressedfilename.tgz directoryname
```

Combining Commands

- Pipes

```
matthew@seymour:~$ ps aux | grep nethack
```

```
matthew@seymour:~$ ps aux | grep nethack | wc –l
matthew@seymour:~$ ps aux --sort=-%cpu | grep -v `whoami`
```

```
matthew@seymour:~$ dpkg --get-selections | grep ftp | sort
```

Combining Commands

- Combining Commands with Boolean Operators
  - The && operator, when added to the end of a command, reads that exit status and confirms its value as 0 for true before allowing the next command to be run

```
matthew@seymour:~$ i && k
```

  - The operator ||, which runs the following command only if the first one returns an exit status of 1 for false

```
matthew@seymour:~$ m || n
```

Combining Commands

- Process Substitution
  - the output of one or more commands is precisely what you want to use as the input to another command

matthew@seymour:~$ **more <(ls -al)**

No space

matthew@seymour:~$ **diff <(ls** *firstdirectory***) <(ls** *seconddirectory***)**

Using Environment Variables

- PWD—Provides the full path to your current working directory, used by the pwd command, such as /home/matthew/Documents
- USER—Declares the user's name, such as matthew
- LANG—Sets the default language, such as English
- SHELL—Declares the name and location of the current shell, such as /bin/bash
- PATH—Sets the default locations of executable files, such as /bin, /usr/bin, and so on
- TERM—Sets the type of terminal in use, such as vt100, which can be important when using screen-oriented programs, such as text editors
- ……

You can print the current value of any environment variable by using matthew@seymour:~**$ echo *$VARIABLENAME***

You can use the **env** or **printenv** command to display all environment variables

Exercise

- Download and install **Eclipse** on Ubuntu
  - Check for Java installed,
  - Check OS type (64/32bit)
  - Download Eclipse tarball file from
    - https://www.eclipse.org/downloads/eclipse-packages/
  - Extract and install
    - tar –xf <tarball file.gz>
  - Run Eclipse
  - Set PATH to run from $HOME directory
    - PATH = $PATH:$HOME/<install directory>

Note: to permanently set PATH env, edit file $HOME/.profile

QUESTION
&
ANSWER