UNIX Fundamentals Part II

Presented By:

Gerry Hounsell

UNIX Fundamentals
COURSE AGENDA PART I

- UNIX History
- The Many Flavours' of UNIX
- The Structure of UNIX
- Access to UNIX Systems
- Processes
- Filesystems & Directories
- Devices

UNIX Fundamentals COURSE AGENDA PART II

- ❑ vi Editor
- ❑ Redirection & Regular Expressions
- ❑ Scheduling
- ❑ Logical Volume Management
- ❑ Printing
- ❑ UNIX Tools & Utilities
- ❑ System Error Reporting
- ❑ Manual Pages

UNIX Fundamentals Part II

- vi editor
- Redirection & Regular Expressions
- Scheduling
- Logical Volume Management
- Printing
- UNIX Tools & Utilities
- System Error Reporting
- Manual Pages

vi editor

- What is vi?
- A brief History of vi
- Invoking vi
- vi – command & input modes
- Moving around in vi
- Inserting, Deleting & Saving
- Cutting & Pasting
- Find & Search
- External Files
- Substitution

vi editor – What is vi?

- vi is a Visual Editor (hence the name -- vi for VIsual).
- vi is default visual editor under Unix, and is therefore shipped with all versions of Unix.
- vi is a modal editor and assigns different meanings to buttons or keystrokes depending on the active editing mode.
- There are three modes for vi:
  - Command
  - Text entry or Input
  - Vi mode
- vi is case-sensitive: J and j do very different things!

vi editor - A Brief History of vi

- A line Editor was developed at Bell Labs for the early version of AT&T UNIX. It was called ed.
- Many other editors were 'spawned' from ed. These included:
  - edlin (DOS editor)
  - sed (non-interactve stream editor – UNIX)
  - ex (BSD extended version of ed)
- Bill Joy at University of California, Berkeley, created the extended version of ed (ex) around the time that terminal devices (cathode ray tube screen display) were being introduced (1970's).
- Later he added **v**isual **i**nterface mode to ed, which was invoked by using the command **:vi** <return>. Later, a linked filename was created to invoke the ex editor in visual mode. The name of the link was: **vi.**

vi editor – Command Input & vi modes

- There are three operational modes to the vi editor:

  - Vi mode:
    - ***This is the mode vi starts in.***
    - ***Most keys on the keyboard are defined to be a specific command.***
    - ***As the key or key Sequence is issued, that command is executed.***
    - ***At any time, pressing the <ESC> key returns the user to vi mode.***

  - Command mode:
    - ***to reach this mode, you MUST first be in vi mode***
    - ***Issue a colon (``:") to enter the command mode***
    - ***The colon will appear at the bottom left corner of the screen.***
    - ***The command may be issued following the colon.***

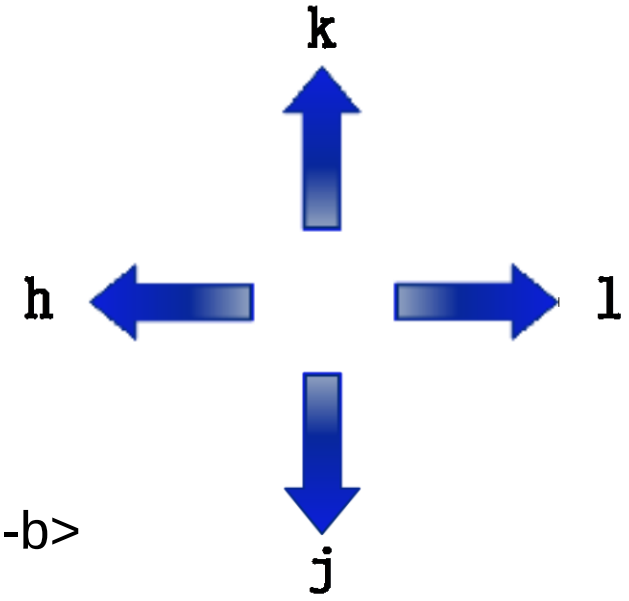
  - Input mode:
    - ***this is where most users expect an editor to start.***
    - ***This "mode" actually refers to commands issued from vi mode but that allow the user to start inputting data into the file.***

vi editor – Invoking & Closing vi

- vi <filename> Open file for editing.
- vi +n <file> Open file and go to line n.
- vi -r <file> Recover a failed editing session.
- To close a vi session:
  - Without saving file `:q!`
  - Saving file `:wq`

vi editor – Moving Around In vi

- To move around in the file, use the arrow keys.
- If number is typed immediately before pressing the arrow key, the position of the cursor will move number positions in the direction of the arrow.
- For terminals with no functional arrow keys, these four keys will move the cursor around h, j, k, l:



- Forward One Screen use ^f <CTRL-f>
- Backwards One Screen use ^b <CTRL-b>
- End of File G <SHIFT G>

vi editor – Inserting, Deleting & Saving

- Inserting data
  - **i** Insert Mode

- Deleting data
  - **x** OR **d** Delete Character
  - **dd** Delete Line
  - **[num] dd** will delete num lines beginning at the current line. The default num is 1.
  - **D** deletes until the end of the line.

- Saving File(s)
  - **:w** save (don't quit) [:w filename saves to filename]
  - **:q** save and quit
  - **:q!** quit (don't save)
  - **:x** save if a change has been made, quit regardless
  - **ZZ** same as :x

vi editor – Copying & Pasting

- Copying
  - **yy** Copy Line in buffer
  - **[num] yy** will copy num lines beginning at the current line, into a buffer. The default num is 1.

- Pasting
  - **p** Put Buffer (When a line is deleted with dd (num dd), these lines are copied into a special buffer. **p** will put that buffer after the current line).
  - Note that the contents of that buffer are not erased; they can be put (recovered) as many times as desired.

- Undo
  - **u** Undo
  - This is a very useful command. It cancels the effect of the previously executed command.
  - vim, a vi clone used mosly on linux distibutions, has an unlimitted undo.
  - To redo the undone change, use **CTL-R**.

vi editor – Find/Search

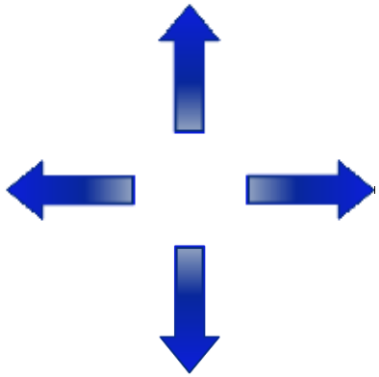
- Search for a pattern:
  - ***/pattern*** find the first occurrence of *string* after the cursor.
  - **?pattern** find the first occurrence of *string* before the cursor.
  - **n** repeat search in same direction
  - **N** reverse direction
  - **/** Repeat previous search forwards
  - **?** Repeat previous search backwards

- Search for a character
  - **f[x]** Search for character x in current line forwards
  - **F[x]** Search for character x in current line backwards
  - **;** Repeat last current-line search in same direction
  - **,** Repeat last current-line search in opposite direction

Vi Editor Checkpoint - 1

1. Which four keys are used to move the cursor by one position within a vi editor session and which key moves the cursor in which direction?

____ ____ ____ ____



Vi Editor Checkpoint - 2

2. Name two different commands to exit from the vi editor:

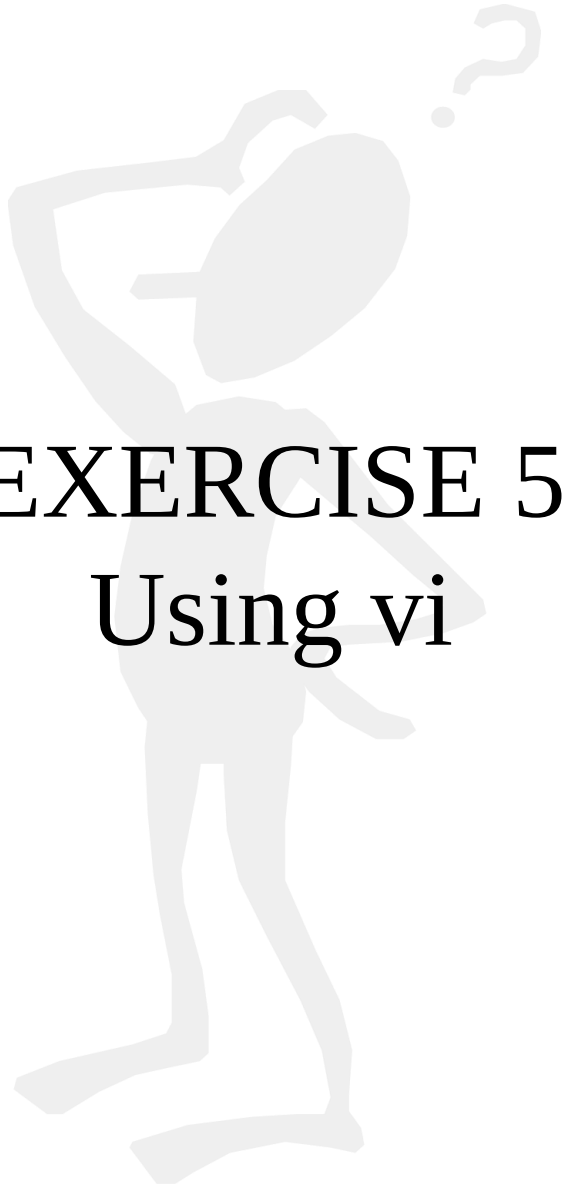
   _____ _____

3. Which command can you use to move 5 lines down? _____

4. Give the command to move to line number 35? _____

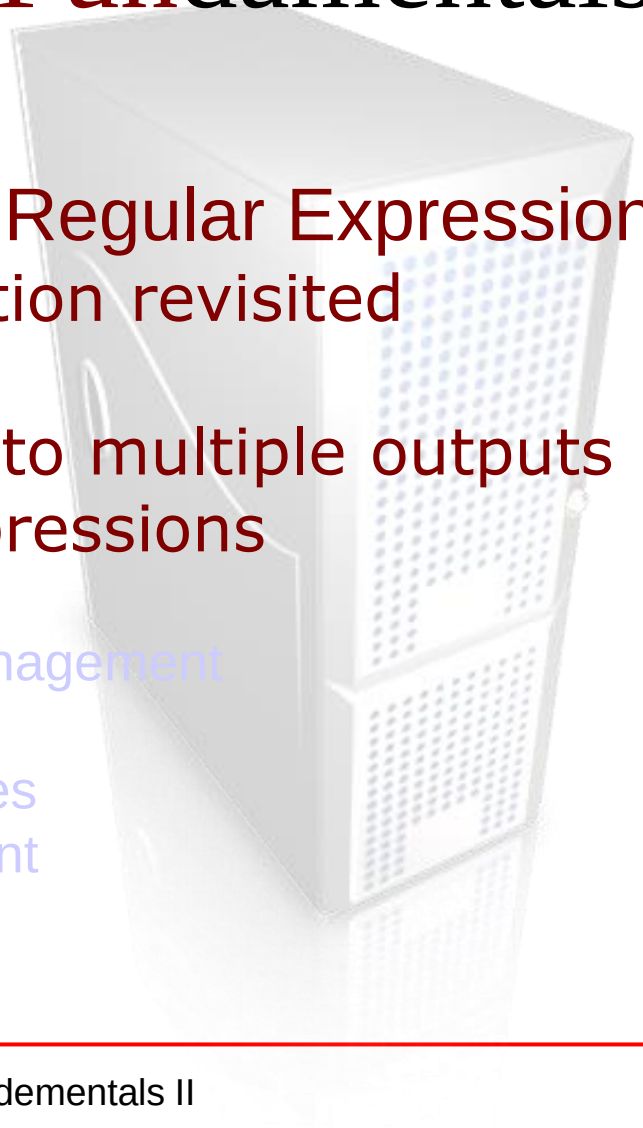
5. How would copy the current line to the copy/paste buffer? _____

6. What command can you use to goto line 1 in a file? _____



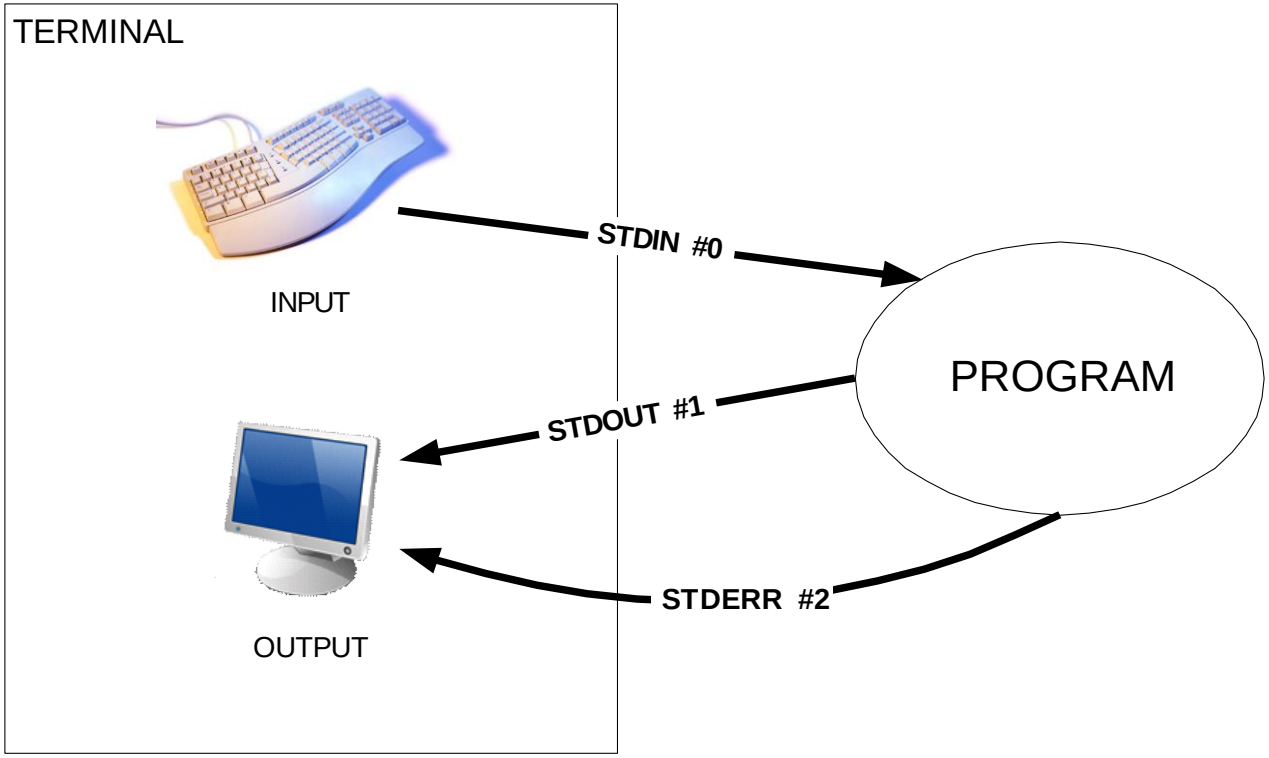
EXERCISE 5
Using vi

UNIX Fundamentals Part II



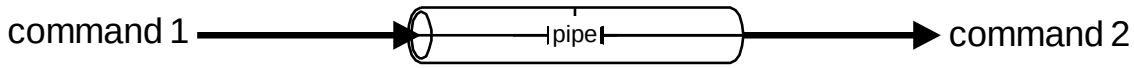
- vi editor
- Redirection & Regular Expressions
  - I/O Redirection revisited
  - Pipes
  - Redirection to multiple outputs
  - Regular Expressions
- Scheduling
- Logical Volume Management
- Printing
- UNIX Tools & Utilities
- System Management
- Manual Pages

I/O Redirection Revisited



Pipes

- Programs can be run together such that one program reads the output from another with no need for an explicit intermediate file:



- Example:
  - Command 1 | command 2
  - ls | grep '.sh'

Redirection to multiple outputs

- The tee command is used to write to the standard output and to a file simultaneously.
- The tee command anywhere in a pipe command to divert a copy of the standard input (of tee) to disk and another copy to a terminal.
- Name is from the world of plumbing; it allows one input and two outputs. (tee command actually allows multiple file copies).
- Example:

  - `Date | tee -a job.log`

Regular Expressions

- A regular expression is a sequence or a pattern of characters that is matched against a string of text when performing searches and replacements.
- Regular expressions, are used in pattern matching and substitution operators.
- Simple regular expression consists of a single character or a set of characters that matches itself.
- Regular expressions are very powerful tool. Packing a lot of meaning into a short space.
- Many tasks can be carried out with regular expressions.
  - The most common use is to find out whether a given string matches a particular pattern.
  - Can be used to find out where the matching substring is located within the string.
  - Can also use a substitution command to replace matching sections
- Examples:

*	all files
.	files with a .
*.c	files that end with .c
*.?	.c, .h, .o, ... files
.[A-Z]*	.Xdefaults, ...
*.[ch]	files ending in .c or .h

Regular Expressions I

- Regular expression examples:
  - ls -l | grep ^d - lists directories only in the long ls listing.
  - ls [A-Z]* - lists files with names starting in caps.
  - ls -al | grep -v ^[A-Z] - lists files with names which do not begin in caps.
  - cat midsummer

I know a bank where the wild thyme blows,
Where oxlips and the nodding violet grows,
Quite over-canopied with luscious woodbine,
With sweet musk-roses and with eglantine.

  - grep [a-z] midsummer

I know a bank where the wild thyme blows,
Where oxlips and the nodding violet grows,
Quite over-canopied with luscious woodbine,
With sweet musk-roses and with eglantine.

More on grep later!

Input/Output Redirection Checkpoint I

1. Name the three file descriptors assigned by the shell when a program starts?

   i) ..... ii) ..... iii) .....

2. How can we redirect the output of the ps command to a file?

   _____

3. How can we ‘pipe’ the output of the ps command to the pg command?

   _____

Input/Output Redirection Checkpoint II

4. How would you search/list ALL files ending in .c or .h

   _____

5. How would you match any character except uppercase A, B, C, D, or E.

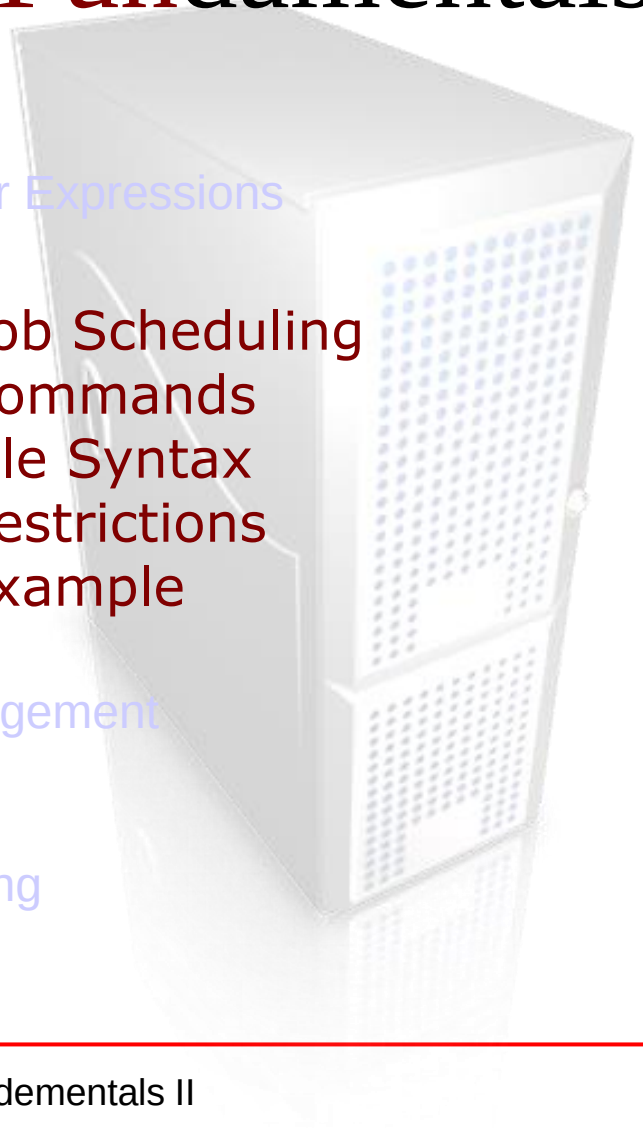
   _____

6. How would you match ^m in a file:

   _____

EXERCISE 6
Redirection & Regular Expressions

UNIX Fundamentals Part II



- vi editor
- Redirection & Regular Expressions
- Scheduling
  - Overview of Job Scheduling
  - CRONTAB – Commands
  - CRONTAB – File Syntax
  - CRONTAB – Restrictions
  - CRONTAB – Example
  - at command
- Logical Volume Management
- Printing
- UNIX Tools & Utilities
- System Error Reporting
- Manual Pages

Scheduling – CRONTAB Overview

- CRONTAB/CRON is the UNIX scheduler and is used to run commands at a later time.
- CRONTAB is used to automate repetitious jobs.
- CRON (daemon) is driven by CRONTAB, a configuration file that specifies shell commands to run periodically on a given schedule.
- CRONTAB command submits, edits, lists, or removes cron jobs.
- All users CAN have their own CRONTAB (depends on security/permissions settings)
- Users can only create/edit/use their own CRONTAB files (except for root!)

Scheduling – CRONTAB Commands

- CRONTAB commands:
  - crontab -e Edit your crontab file, or create one if it doesn't already exist.
  - crontab -l Display your crontab file.
  - crontab -r Remove your crontab file.
  - crontab -v Display the last time you edited your crontab file. (This option is only available on a few systems.)

Scheduling – CRONTAB File Syntax

- A crontab file has five fields for specifying day , date and time followed by the command to be run at that interval.

```
*    *    *    *     *  command to be executed
-    -    -    -     -
|    |    |    |     |
|    |    |    |     +----- day of week (0 - 6) (Sunday=0)
|    |    |    +------- month (1 - 12)
|    |    +--------- day of month (1 - 31)
|    +------------ hour (0 - 23)
+-------------- min (0 - 59)
```

- * in the value field above means all legal values as in braces for that column. The value column can have a * or a list of elements separated by commas. An element is either a number in the ranges shown above or two numbers in the range separated by a hyphen (meaning an inclusive range).

- Example:

```
30     18     *     *     *          rm /home/someuser/tmp/*
```

Scheduling – CRONTAB File Syntax I

- A line in crontab file like below removes the tmp files from /home/someuser/tmp each day at 6:30 PM.

```
30    18    *    *    *        rm /home/someuser/tmp/*
```

- Changing the parameter values as below will cause this command to run at different time schedule below :

Minute	Hour	Day/Month	Month	Day/Week	Execution Time
30	0	1	1,6,12	*	- - 00:30 Hours on 1st January, June & December
0	20	*	10	1-5	- - 8:00pm every weekday (Mon-Fr) only in October
0	0	1,10,15	*	*	- - Midnight on 1st, 10th & 15th of month.
5,10	0	10	*	1	- - At 12:05, 12:10 every Monday & 1oth of every month.

Scheduling – CRONTAB Restrictions

- A user can execute crontab if their name appears in the file **cron.allow**.
- If the user does not exist in the **cron.allow** file, they can use crontab if their name does not appear in the file **cron.deny**.
- If only **cron.deny** exists and is empty, all users can use crontab. If neither file exists, only the root user can use crontab.
- The allow/deny files consist of one user name per line.
- The allow/deny files exist in different locations for different flavours of UNIX, examples include:
  - /usr/lib/cron/ (Solaris)
  - /etc/ (Linux)
  - /var/adm/cron/ (AIX)

Scheduling – CRONTAB File Syntax II

- Disable eMail
  - By default cron jobs sends a email to the user account executing the cronjob. If this is not needed put the following command At the end of the cron job line.

```
>/dev/null 2>&1
```

- Generate log file
  - To collect the cron execution in a log file :

```
30 18 * * * rm /home/someuser/tmp/* > /home/someuser/cronlogs/clean_tmp_dir.log
```

Scheduling – at command

- The **at** command is used to schedule commands to be executed once at a particular time in the future.
- The at command is normally used for ‘one-off’ scheduled jobs, as an alternative to cron.
- The at-job inherits the current environment (variables, etc) which can be useful.
- The **atq** command displays the current user's queue of jobs that are waiting to be run at a later date, sorted in the order the jobs will be run.
- The **atrm** command removes jobs that were created with the at command, but have not executed.

Scheduling – at command I

- Examples:
  - To run the killicon.ksh script at 3:00 in the afternoon on the 24th of January, enter any one of the following commands:

    - `echo killicon.ksh | at 3:00 pm January 24`
    - `echo killicon.ksh | at 3 pm Jan 24`
    - `echo killicon.ksh | at 1500 jan 24`

  - In order to look at the queue created by the at command, enter: **atq** If there are jobs in the queue, a message similar to the following appears:

```
root.635623200.a          Wed     Feb 21  12:00:00 1990
root.635670000.a          Thu     Feb 22  01:00:00 1990
```

  - To remove job number root.62169200.a from the at command queue, enter:

```
atrm root.621619200.a
```

Scheduling Checkpoint

1. What daemon runs the crontab command?

   _____

2. How do you control which users are allowed to use the crontab command?

   _____

3. How do you list the contents of a crontab file?

   _____

Scheduling Checkpoint I

4. What command would a user run to check the jobs they had scheduled to run once?

_____

5. How would you stop emails being generated as cron jobs are run?

_____

6. How would you log cron job execution and where would you configure this?

_____

7. Where are users crontab files located?

_____

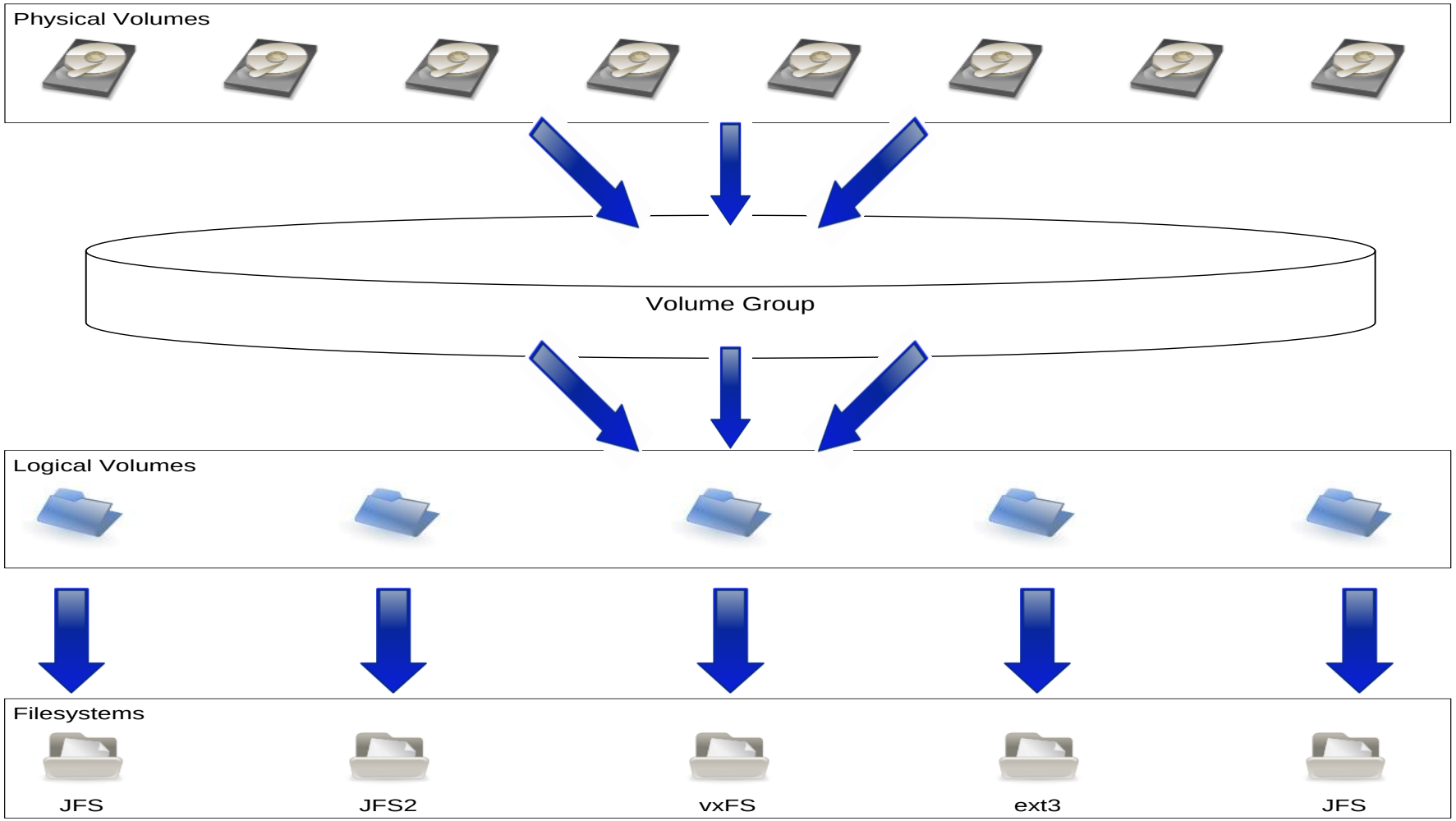
UNIX Fundamentals Part II

- vi editor
- Input/Output Redirection
- Scheduling
- **Logical Volume Management**
  - Volume Managers
  - Basic Concepts & Terminology
  - Disk Utilities
- Printing
- UNIX Tools & Utilities
- System Management
- Manual Pages

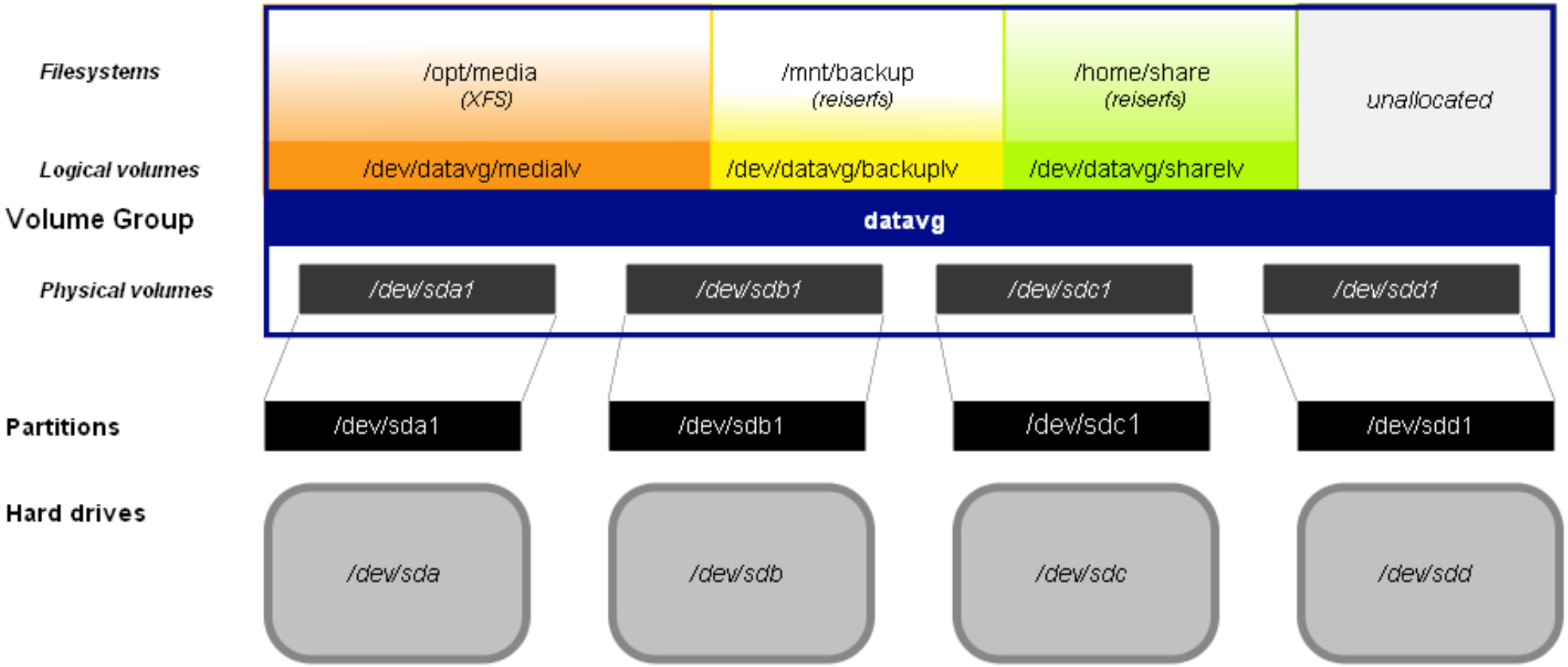
LVM – Volume Managers

- Most modern UNIX variants use a Volume Manager to abstract the file-system away from actual physical devices.
- Why?
  - Group disks together for resilience or performance
  - Grow over traditional disk boundaries
  - use and allocate disk space more efficiently and flexibly
  - move logical volumes between different physical devices
  - have very large logical volumes span a number of physical devices
  - take snapshots of whole filesystems easily, allowing on-line backup of those filesystems
  - replace on-line drives without interrupting services

LVM – Basic Concepts & Terminology



LVM – Linux Example



LVM – Disk Utilities

- **df** - (abbreviated from **d**isk **f**ree) displays information about total space and available space on a file system.
  - FileSystem parameter specifies the name of the device on which the file system resides, the directory on which the file system is mounted, or the relative path name of a file system.
  - The File parameter displays information for the file system on which the file or directory resides.
  - No parameter displays information for all currently mounted file systems.

- df example

```
srublba01:root:/> df -k
Filesystem 512-blocks Free   %Used   Iused  %Iused  Mounted on
 /dev/hd0    19368     9976    48%     4714    5%     /
 /dev/hd1    24212     4808    80%     5031   19%     /usr
 /dev/hd2     9744     9352     4%     1900    4%     /site
 /dev/hd3     3868     3856     0%      986    0%     /usr/venus
```

LVM – Disk Utilities I

- **du** - (abbreviated from **d**isk **u**sage) is used to display the file space usage; space used under a particular directory or files on a file system.
  - If the File parameter specified is actually a directory, all files within the directory are reported on.
  - If no File parameter is provided, the du command uses the files in the current directory.

- du examples

```
srublba01:root:/opt/nmon/bin> du -sk
26180.
srublba01:root:/opt/nmon/bin> du
52360   .
srublba01:root:/opt/nmon/bin> du -sk C1430099.tar.backup
5512    C1430099.tar.backup
```

LVM – Disk Utilities II

- **mount** - Makes a filesystem ready for use.

- mount example

```
srublba01:root:/> mount
  node       mounted        mounted over    vfs       date        options
-------- ---------------  ---------------  ------ ------------ ---------------
         /dev/hd4         /              jfs    11 Feb 07:47 rw,log=/dev/hd8
         /dev/hd2         /usr             jfs    11 Feb 07:47 rw,log=/dev/hd8
         /dev/hd9var      /var             jfs    11 Feb 07:47 rw,log=/dev/hd8
         /dev/hd3         /tmp             jfs    11 Feb 07:47 rw,log=/dev/hd8
         /dev/hd1         /home            jfs    11 Feb 07:48 rw,log=/dev/hd8
         /proc            /proc            procfs 11 Feb 07:48 rw
         /dev/hd10opt     /opt             jfs    11 Feb 07:48 rw,log=/dev/hd8
         /dev/varloglv    /var/log         jfs    11 Feb 07:48 rw,log=/dev/hd8
         /dev/hsbclv      /hsbc            jfs    11 Feb 07:48 rw,log=/dev/hd8
         /dev/auditlv     /audit           jfs    11 Feb 07:48 rw,log=/dev/hd8
```

Logical Volume Management Checkpoint I

1. Give 3 reasons why we use Logical Volume Manager's?

2. What are Physical Volumes (PV's) divided into?

Logical Volume Management Checkpoint II

3. Volume Groups contain _____ and _____.

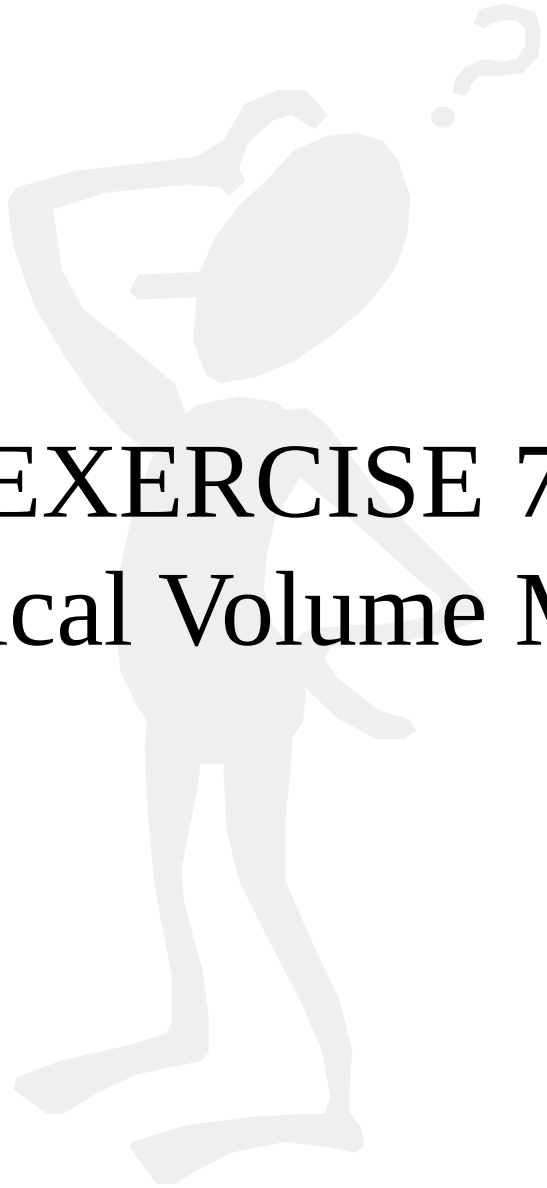
4. What commands would you use to check disk/filesystem capacity

_____

_____

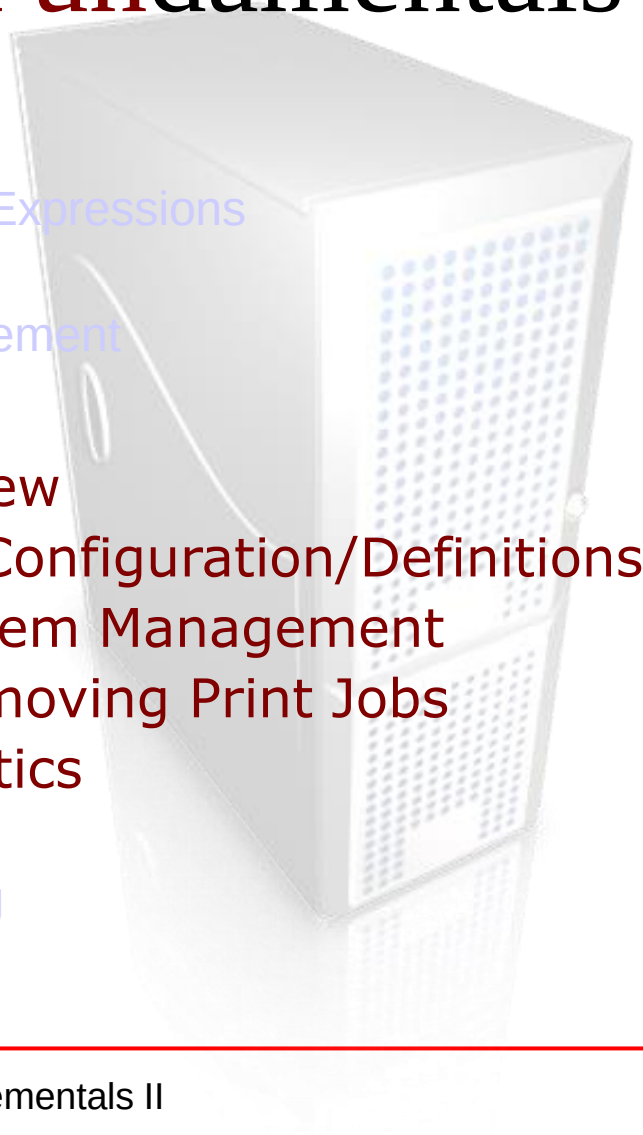
5. What command do you issue to make a filesystem available for use?

_____



EXERCISE 7
Disk & Logical Volume Management

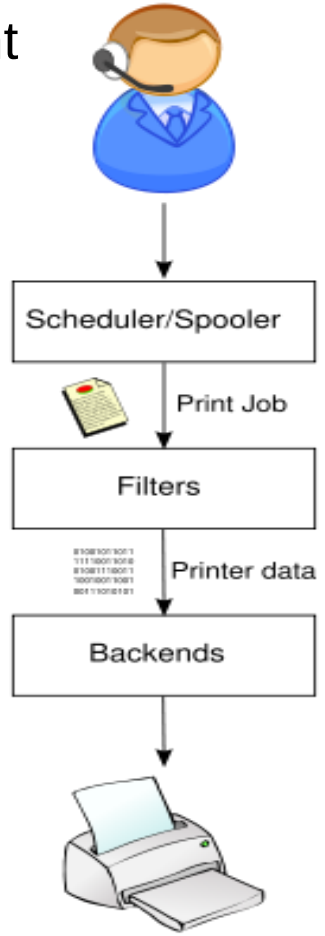
UNIX Fundamentals Part II



- vi editor
- Redirection & Regular Expressions
- Scheduling
- Logical Volume Management
- Printing
  - Printing Overview
  - Printer Queue Configuration/Definitions
  - Printer Subsystem Management
  - Submitting/Removing Print Jobs
  - Print Job Statistics
- UNIX Tools & Utilities
- System Error Reporting
- Manual Pages

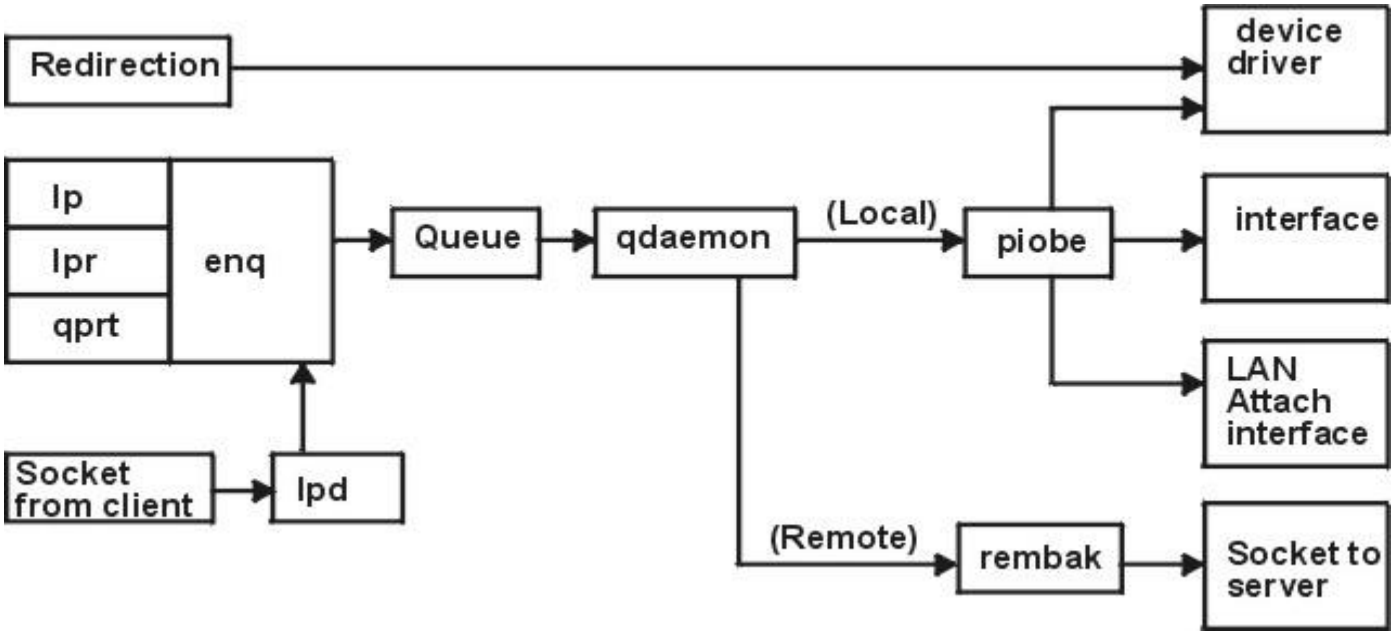
Printing - Overview

- A number of different print commands & systems – dependent on flavour of UNIX
- UNIX command line printing programs, assume that the default print queue name is lp.
- 4 main printing systems in UNIX/Linux:
  - System V printing system (lpr)
  - Berkely Printing System (lpd)
  - Line Printer Remote next generation (LPRng)
  - Common UNIX Printing System (CUPS)



Printing – Overview I

- AIX Print Subsystem example:



Printing – Queue Definitions

- Printer queue Definition's:

  - AIX
    `/etc/qconfig`
  - HP-UX
    `/etc/lp/interface/*`
  - SUN Solaris
    `/etc/lp/interface/*`
  - Linux
    `/var/spool/lpd/lp/*`
  - BSD
    `/var/spool/print`

Printing – Start/Stop Print Subsystem

- Starting/Stopping the Printing Subsystem:

  - AIX
    Start: `startsrc -s lpd` **Stop:** `stopsrc -s lpd`
  - HP-UX
    Start: `lpsched` **Stop:** `lpshut`
  - SUN Solaris
    Start: `/usr/lib/lp/lpsched` **Stop:** `/usr/lib/lp/lpshut`
  - Linux
    Start: `/etc/init.d/lpd start` **Stop:** `/etc/init.d/lpd stop`
  - BSD
    Start: `lpd` **Stop:** `lpd`

Printing – Submitting Print Jobs

- Submitting Print Jobs:
  - AIX
    ```
    enq OR lp OR lpr OR qprt
    ```
  - HP-UX
    ```
    lp
    ```
  - SUN Solaris
    ```
    lp OR lpr
    ```
  - Linux
    ```
    lpr
    ```
  - BSD
    ```
    lp
    ```

Printing – Removing Print Jobs

- Removing Print Jobs:
  - AIX
    ```
    cancel OR lprm OR qcan OR enq -x
    ```
  - HP-UX
    ```
    cancel
    ```
  - SUN Solaris
    ```
    cancel OR lprm
    ```
  - Linux
    ```
    lprm
    ```
  - BSD
    ```
    cancel OR lprm
    ```

Printing – Print Job Statistics

- Print Queue/Job Statistics:
  - AIX

    ```
    enq -A OR lpq OR lpstat OR qchk
    ```

  - HP-UX

    ```
    lpstat
    ```

  - SUN Solaris

    ```
    lsptat
    ```

  - Linux

    ```
    lpq
    ```

  - BSD

    ```
    lpq
    ```

Printing Checkpoint I

1. Where are the print queue definitions kept for the following flavours of UNIX:

   a) AIX _____

   b) Linux _____

   c) Solaris _____

   d) HP-UX _____

2. What process listens for users print requests?

   _____

Printing Checkpoint II

3. Which command would you use to cancel print jobs for the following flavours of UNIX:

   a) AIX _____

   b) Linux _____

   c) Solaris _____

   d) HP-UX _____

4. How would you view the print queue statistics in AIX & Linux?

   _____

UNIX Fundamentals Part II

- vi editor
- Redirection & Regular Expressions
- Scheduling
- Logical Volume Management
- Printing
- **UNIX Tools & Utilities**
- System Error Reporting
- Manual Pages

UNIX Tools & Utilities

- find
- grep
- sort
- head
- tail
- Transferring DOS Data Files
- diff
- cmp
- dircmp
- file
- alias
- which
- Whereis
- tr
- File Compression

UNIX Tools & Utilities - find

- The find command is used to locate files on a Unix or Linux system.
- find will search any set of directories specified for files that match the supplied search criteria.
- Search can be for files by name, owner, group, type, permissions, date, and other criteria.
- The search is recursive in that it will search all subdirectories too.
- Find command syntax:
  - find <path> <expression> <action>

UNIX Tools & Utilities – find I

- Examples of find command:
  - Search for file with a specific name in a set of files (-name)
    - ***find . -name "rc.conf" -print***
    - ***This command will search in the current directory and all sub directories for a file named rc.conf.***
    - ***Note: The -print option will print out the path of any file that is found with that name. In general -print wil print out the path of any file that meets the find criteria.***
  - How to apply a unix command to a set of files (-exec).
    - ***find . -name "rc.conf" -exec chmod o+r '{}' \;***
    - ***This command will search in the current directory and all sub directories. All files named rc.conf will be processed by the chmod -o+r command. The argument '{}' inserts each found file into the chmod command line. The \; argument indicates the exec command line has ended.***
    - ***The end results of this command is all rc.conf files have the other permissions set to read access (if the operator is the owner of the file).***

UNIX Tools & Utilities – find II

- Search for any occurrence of nmon:

```
srublba01:root:/> find / -name nmon -print
/home/baconr/nmon
/home/greavesc/nmon
/opt/nmon
/opt/nmon/bin/nmon
/opt/wasinf/utils/shScript/nmon
/usr/bin/nmon
```

- Search for all .csv files that have been changed in the current 24 hour period:

```
gbsrual0048:root:/opt/nmon/bin> find /var -name "*.csv" -ctime 1 -print
/var/nmon/pdb2b_gbsrual0048.irssl01.HFM1.IBMDEFAULTBP_d_070917.csv
/var/nmon/pdb2_gbsrual0048.irssl01.HFM1_d_070916.csv
/var/nmon/netp_gbsrual0048_d_070918.csv
/var/nmon/pdb2_gbsrual0048.irssl01.HFM1_d_070917.csv
/var/nmon/pdb2b_gbsrual0048.irssl02.HFMFB.IBMDEFAULTBP_d_070916.csv
/var/nmon/pdb2b_gbsrual0048.irssl01.HFM1.IBMDEFAULTBP_d_070916.csv
/var/nmon/pdb2b_gbsrual0048.irssl02.HFMFB.IBMDEFAULTBP_d_070917.csv
```

UNIX Tools & Utilities - grep

- grep searches for lines of text that match one or many regular expressions, and outputs only the matching lines.
- grep syntax:

```
grep [options] pattern [file1 file2 ..]
```

- grep can be used with regular expressions
  - Patterns with metacharacters should be in single quotes (‘ ’). This ensure the shell ignores them
  - Valid metachracters for use with grep:
    - ***. Any single character***
    - **** Zero or more occurrences of the preceeding character***
    - ***[a-f] Any one of the characters in the range of a through z.***
    - ***^a Any lines that start with an a.***
    - ***z$ Any lines that end with a z.***

UNIX Tools & Utilities – grep I

- Example 1. Search for the root entry in the password file:

```
srublba01:root:/> grep root /etc/passwd
root:!:0:0::/:/usr/bin/ksh
srublba01:root:/>
```

- Example 2 – Search for processes belonging to user patrolag:

```
srublba01:root:/> ps -ef | grep patrolag
patrolag  528390       1   0   11 Feb      -  0:00 ds_listener -port=50005 -hosts= -period=3 -interval=1440 -
        log=/opt/patrol/dsclient/log/dslistener.log –start
patrolag 1065096       1   0   12 Feb      - 914:53 /usr/adm/best1_default/bgs/bin/bgsagent -m MetricTable.mgt -b
        /usr/adm/best1_default
patrolag 1314930 1609924   0   12 Feb      - 13:35 /opt/patrol/PATROL/AIX5.3-64/../pmg/AIX5.3-64/pmgreader
patrolag 1609924       1   0   12 Feb      - 3444:18 PatrolAgent -p 3300
patrolag 2285774 1065096   1   12 Feb      - 13975:32 bgscollect -I noInstance -B /usr/adm/best1_default
patrolag 3035308 1609924   0   12 Feb      - 500:59 /opt/patrol/PATROL/AIX5.3.0.0-64/best1/7.2.10/bgs/bin/dcm -f
        /opt/patrol/PATROL/AIX5.3-64/log/patrol.FIFO-srublba01-3300
patrolag 3174428 1609924   0   12 Feb      -  0:59 /opt/patrol/PATROL/AIX5.3-64/../pmg/AIX5.3-64/pmgpipereader
root 3301446 1138876   0 16:46:20  pts/3  0:00 grep patrolag
srublba01:root:/>
```

UNIX Tools & Utilities - sort

- The sort command is used to sort files, merge files that are already sorted, and check files to determine if they have been sorted.
- The sort command requires a temporary area to process data. This can be one of the following filesystems:
  - /var/tmp Temporary space during the sort command processing.
  - /usr/tmp Temporary space during the sort command processing, if file cannot be created in /var/tmp.
  - /tmp Temporary space during the sort command processing, if file cannot be created in /var/tmp or /usr/tmp.

UNIX Tools & Utilities – sort I

- Example 1:

```
srublba01:root:/opt/nmon/bin> ls -s | sort -n
132 lsof
 132 nmon32
 332 nmon_aix51
 376 nmon_aix52ml5
 380 nmon_aix52ml2
 396 nmon_aix53
 396 nmon_aix53.11
1900 srublba01.html
5512 C1430099.tar.backup
15756 core
```

- Exampe 2:

```
srublba01:root:/opt/nmon/bin>  du /bin/* | sort -nr| pg
1544    /bin/rksh93
1544    /bin/ksh93
1408    /bin/X11r5
1400    /bin/X11r5/Motif1.2
1264    /bin/bsh
1000    /bin/gdam
848     /bin/sftp-server2
824     /bin/ssh-certview
808     /bin/smitty
760     /bin/xhconx
```

UNIX Tools & Utilities - head

- Displays the first few lines or bytes of a file, files or piped output.
- If no flag is specified with the head command, the first 10 lines are displayed by default.
- The File parameter specifies the names of the input files.
- An input file must be a text file.
- When more than one file is specified, the start of each file will look like the following:

```
==> filename <==
```

UNIX Tools & Utilities – head I

- Example 1:

```
srublba01:root:/opt/nmon/bin> head readme.txt
README.txt for nmon

nmon is a free performance tool for AIX and Linux available from
http://www.ibm.com/developerworks/eserver/articles/analyze_aix/index.html

There is also a spreadsheet analyser for nmon captured data from
http://www.ibm.com/developerworks/eserver/articles/nmon_analyser/index.html

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Remember: The nmon tool is NOT OFFICIALLY SUPPORTED.
srublba01:root:/opt/nmon/bin>
```

- Example 2:

```
srublba01:root:/opt/nmon/bin> head -n 4 readme.txt
README.txt for nmon

nmon is a free performance tool for AIX and Linux available from
http://www.ibm.com/developerworks/eserver/articles/analyze_aix/index.html
```

UNIX Tools & Utilities - tail

- The tail command is used to display the last few lines of a text file or piped data.
- By default, tail will print the last 10 lines of its input to the standard output.
- With command line options the number of lines printed and the printing units (lines, blocks or bytes) may be changed.
- tail has a special command line option -f (follow) that allows a file to be monitored.

UNIX Tools & Utilities – tail I

- Example 1:

```
srublba01:root:/opt/nmon/bin> tail -n 8 readme.txt
e) AIX 5.2 capture vmo, ioo, no, nfso and schedo parameters in full.
f) Remove order dependency in -f, -F, -T -t options.

g) Bug fixes when running on AIX 5.2 (virtual memory stats and multipath I/O).

h) Removed the disk read and write size data - now merged into blocksize
   stats.

--- The End ---
```

- Example 2:

```
Jun 18 18:51:00 srubsrublba01:root:/var/log> tail -f syslog
Jun 20 15:51:00 srublba01 mail:info sendmail[3670062]: l5KEp0d2056256:
   to=root, ctladdr=daemon (1/1), delay=00:00:00, xdelay=00:00:00,
   mailer=local, pri=120662, dsn=2.0.0, stat=Sent
Jun 20 15:51:04 srublba01 auth|security:notice last message repeated 2 times
Jun 20 15:52:01 srublba01 user:info syslog: /usr/sbin/ifconfig en1
Jun 20 15:52:01 srublba01 user:info syslog: /usr/sbin/ifconfig -a
Jun 20 15:52:03 srublba01 auth|security:notice su: from root to root at
   /dev/tty??
```

UNIX Tools & Utilities – DOS/UNIX I

- convert a UNIX text file to DOS format
  - AIX
    - ***doswrite [-a] [-v] [-DDevice] File1 File2***
    - ***command converts lowercase characters specified in the File1 parameter to uppercase***
    - ***all file names are assumed to be full (not relative) path names, you do not need to add the initial / (slash).***
  - Other flavours of UNIX
    - ***unix2dos [-ascii] [-iso] [-7] [-437 | -850 | -860 | -863 | -865] originalfile convertedfile***
    - ***The unix2dos utility converts ISO standard characters to the corresponding characters in the DOS extended character set.***
    - ***If the original file and the converted file are the same, dos2unix will rewrite the original file after converting it.***

UNIX Tools & Utilities – DOS/UNIX II

- convert a DOS text file to UNIX format
  - AIX
    - ***dosread** [-a] [-v] [-D Device] File1 [File2]*
    - ***dosread command copies the DOS file specified by the File1 variable to standard output or to the file specified by the File2 variable.***
    - ***dosread command copies the number of bytes specified in the directory entry for the file specified by the File1 variable. Therefore you cannot copy directories because, by convention, directories have a record size of 0.***
  - Other flavours of UNIX
    - ***dos2unix [-ascii] [-iso] [-7] [-437 | -850 | -860 | -863 | -865] originalfile convertedfile***
    - ***dos2unix utility converts characters in the DOS extended character set to the corresponding ISO standard characters.***
    - ***If the original file and the converted file are the same, dos2unix will rewrite the original file after converting it.***

UNIX Tools & Utilities - diff

- The **diff** command compares text files.
- **diff** can compare single files or the contents of directories.
- **Note:** The diff command only works with input files that are text files.
- If the Directory1 and Directory2 parameters are specified, the **diff** command compares the text files that have the same name in both directories. Binary files that differ, common subdirectories, and files that appear in only one directory are listed.

UNIX Tools & Utilities – diff example

- Example:

```
srublba01:root:/var/nmon> diff tunables.before tunables.after
2c2
<       Description = "tunsave -a -F /var/nmon/tunables"
---
>       Description = "tunsave -a -F /var/nmon/tunables.txt"
5c5
<       Last_validation = "2007-03-06 08:57:04 GMT (current, reboot)"
---
>       Last_validation = "2007-03-06 10:31:13 GMT (current, reboot)"
62c62
<       pinnable_frames = "STATIC"      # value was 3877518 (never restored)
---
>       pinnable_frames = "STATIC"      # value was 3876420 (never restored)
```

UNIX Tools & Utilities - cmp

- **cmp** compares two files of any type and writes the results to the standard output.
- By default, cmp is silent if the files are the same.
- If the files differ, the byte and line number at which the first difference occurred is reported.

- Example:

```
gbsrual0048:root:/etc/tunables> cmp lastboot nextboot
lastboot nextboot differ: char 1, line 1
gbsrual0048:root:/etc/tunables>
```

UNIX Tools & Utilities - dircmp

- **dircmp** examines dir1 and dir2 and generates various tabulated information about the contents of the directories.
- Sorted listings of files that are unique to each directory are generated for all the options.
- If no option is entered, a sorted list is output indicating whether the filenames common to both directories have the same contents.

UNIX Tools & Utilities – file

- Determines the file type
- Carries out tests against the file (filelist) to determine type
- The file can be regular file, directory, FIFO(named pipe), block special, character special, symbolic link or sockets type.

- Examples:

```
gbsrual0048:root:/opt/nmon/bin> file p_setup.ksh
p_setup.ksh: shell script  - ksh (Korn shell)
gbsrual0048:root:/opt/nmon/bin> file nmon.tmp
nmon.tmp: empty
gbsrual0048:root:/opt/nmon/bin> file nmon32
nmon32: executable (RISC System/6000) or object module
gbsrual0048:root:/opt/nmon/bin>
```

UNIX Tools & Utilities – alias

- alias is a command that enables a replacement of a word with another string.
- It is mainly used for abbreviating a system command, or for adding default arguments to a regularly used command.
- Typically, an alias will last for the life of the shell session but can be placed in a shell configuration file (~/.cshrc or the systemwide /etc/csh.cshrc for csh)

- Examples:

```
gbsrual0048:root:/> alias
c=clear
cat=/usr/bin/cat
history='fc -l'
integer='typeset -i'
local=typeset
r='fc -e -'
stop='kill -STOP'
```

UNIX Tools & Utilities - which

- **which** - Locates a program file, including aliases and paths.
- For each name given, which searches for the file that would be executed if name were given as a command, and displays the absolute path of that file.
- Each argument is expanded if it is aliased, and searched for along the user's path.
- In the Korn shell, you can use the **whence** command to produce a more verbose report.

- Examples:

```
srublba01:root:/> which nmon
/usr/bin/nmon

$ which hostname
/usr/bin/hostname
```

UNIX Tools & Utilities - whereis

- **whereis** locates source, binary, and manuals sections for specified files.
- The supplied names are first stripped of leading path name components and any (single) trailing extension of the form .ext (such as .c).
- A usage message is returned if a bad option is entered.

- Examples:

```
srublba01:root:/> whereis nmon
nmon: /usr/bin/nmon

srublba01:root:/> whereis man
man: /usr/bin/man /usr/ucb/man /usr/local/man

srublba01:root:/> whereis passwd
passwd: /etc/passwd /usr/bin/passwd
```

UNIX Tools & Utilities - tr

- **tr** (abbreviated from **tr**anslate or **tr**ansliterate) reads from the standard input and writes to the standard output. It takes as parameters two sets of characters, and replaces occurrences of the characters in the first set with the corresponding elements from the other set.
  - Transforming Characters - If String1 and String2 are both specified and the -d flag is not specified, the tr command replaces each character contained in String1 from the standard input with the character in the same position in String2.
  - Deleting Characters Using the -d Flag - If the -d flag is specified, the tr command deletes each character contained in String1 from standard input.
  - Removing Sequences Using the -s Flag - If the -s flag is specified, the tr command removes all but the first character in any sequence of a character string represented in String1 or String2. For each character represented in String1, the tr command removes all but the first occurrence of the character from standard output. For each character represented in String2, the tr command removes all but the first occurrence in a sequence of occurrences of that character in the standard output.

- Examples:
  - To translate braces into parentheses, type:

```
tr '{}' '()' < textfile > newfile
```

This translates each { (left brace) to ((left parenthesis) and each } (right brace) to) (right parenthesis). All other characters remain unchanged.

  - To translate lowercase characters to uppercase, type:

```
tr 'a-z' 'A-Z' < textfile > newfile
```

  - To delete all NULL characters from a file, type:

```
tr -d '\0' < textfile > newfile
```

UNIX Tools & Utilities – Compression

- Compress files:
  - **compress** - AIX
  - **gzip, compress** - Linux
  - **gzip, compress** - Solaris
  - **gzip, compress** - HPUX

- Decompress Files:
  - **uncompress** - AIX
  - **gunzip, bunzip2, uncompress** - Linux
  - **gunzip, bunzip2, uncompress** - Solaris
  - **gunzip, uncompress** - HPUX

- compress Examples:

```
srublba01:root:/var/nmon> compress -v icondead.070530.log
icondead.070530.log: Compression: 74.91% This file is replaced with
    icondead.070530.log.Z.
srublba01:root:/var/nmon>
srublba01:root:/var/nmon> uncompress icondead.070530.log.Z
srublba01:root:/var/nmon>
```

UNIX Tools & Utilities Checkpoint I

1. Give the find command syntax to search for all hidden files in the /tmp directory that were modified less than 5 days ago:

   _____

2. How would you modify all *.conf files to make them world read, write and executable on a system?

   _____

3. How would you check the file /etc/tunables for the existence of the string lru_file_repage?

   _____

UNIX Tools & Utilities Checkpoint II

4. How would you sort the output from du /bin* to display largest first?

_____

5. How would you constantly display the last 10 lines of a log file called syslog?

_____

6. How would you compare the contents of two files named tunables.before and tunables.after?

_____

UNIX Tools & Utilities Checkpoint III

7. Which command would you use to locate the source, binary and manual sections for specified files?

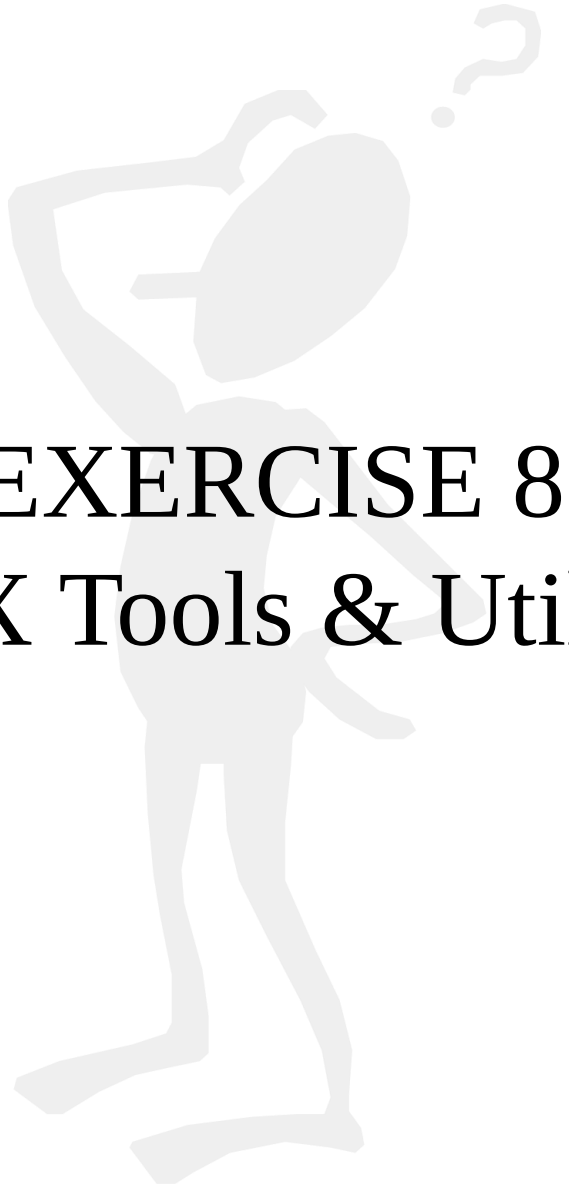
   _____

8. How can you translate lowercase to uppercase?

   _____

9. Name 3 file compression utilities?

   _____ _____ _____



EXERCISE 8
UNIX Tools & Utilities

UNIX Fundamentals Part II

vi editor

- Redirection & Regular Expressions
- Scheduling
- Logical Volume Management
- Printing
- UNIX Tools & Utilities
- **System Error Reporting**
  - Overview
  - Error Log locations
  - Linux Example
  - AIX Example
- Manual Pages

System Error Reporting - Overview

- System error logs hold some important information about the operating system and the hardware.
- In the event of a problem this is always a good place to start.
- System logs reside in different places depending on which flavour of UNIX you are using.

System Error reporting - Log Locations

- Typical log locations include (depending on the UNIX configuration).

  - /var/log (UNIX)
  - /var/adm (AIX)
  - /var/adm/ras (AIX)
  - /var/adm/messages (HP-UX)

System Error Reporting – Linux

- Linux example: The file /var/log/message is the main o/s log file. Using tail –100 will show the last 100 messages.

```
[root@fairmaiden rc5.d]# tail -15 /var/log/messages
Aug 12 13:14:56 fairmaiden nmbd[5569]:   Samba name server FAIRMAIDEN is now a local master browser for workgroup MSHOME
on subnet 192.168.11.5
Aug 12 13:14:56 fairmaiden nmbd[5569]:   *****
Aug 12 13:17:20 fairmaiden smbd[6745]: [2005/08/12 13:17:19, 0] lib/util_sock.c:read_socket_data(384)
Aug 12 13:17:20 fairmaiden smbd[6745]:   read_socket_data: recv failure for 4. Error = Connection reset by peer
Aug 12 13:20:36 fairmaiden sshd(pam_unix)[7012]: session opened for user troyski by (uid=0)
Aug 12 13:20:46 fairmaiden su(pam_unix)[7039]: session opened for user root by troyski(uid=500)
Aug 12 13:24:10 fairmaiden kernel: SMB connection re-established (-5)
Aug 12 14:01:01 fairmaiden crond(pam_unix)[7866]: session opened for user root by (uid=0)
Aug 12 14:01:02 fairmaiden crond(pam_unix)[7866]: session closed for user root
Aug 12 14:26:03 fairmaiden proftpd[8597]: fairmaiden (192.168.11.2[192.168.11.2]) - FTP session opened.
Aug 12 14:26:04 fairmaiden proftpd[8597]: fairmaiden (192.168.11.2[192.168.11.2]) - no such user 'anonymous'
Aug 12 14:26:06 fairmaiden proftpd[8597]: fairmaiden (192.168.11.2[192.168.11.2]) - FTP session closed.
Aug 12 14:26:15 fairmaiden su(pam_unix)[8606]: session opened for user root by troyski(uid=500)
[root@fairmaiden rc5.d]#
```

System Error Reporting – Linux I

- Linux example: The dmesg command shows boot-time and hardware messages.

```
[root@fairmaiden rc5.d]# dmesg
UNIX version 2.6.12-1.1372_FC3 (bhcompile@tweety.build.redhat.com) (gcc vesion 3.4.3
20050227 (Red Hat 3.4.3-22)) #1 Fr
i Jul 15 00:59:10 EDT 2005
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 00000000000a0000 (usable)
 BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
 BIOS-e820: 00000000037f0000 - 00000000037f3000 (ACPI NVS)
 BIOS-e820: 00000000037f3000 - 0000000003800000 (ACPI data)
 BIOS-e820: 00000000fffff0000 - 0000000100000000 (reserved)
 0MB HIGHMEM available.
 55MB LOWMEM available.
Using x86 segment limits to approximate NX protection
On node 0 totalpages: 14320
```

System Error Reporting – AIX I

- AIX system error log example:

```
gbsrual0048:root:/> errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
DE84C4DB   0913160907 I O ConfigRM       IBM.ConfigRM daemon has started.
A2205861   0903222607 P S SYSPROC        Excessive interrupt disablement time
A6DF45AA   0903101807 I O RMCdaemon      The daemon is started.
2BFA76F6   0903101507 T S SYSPROC        SYSTEM SHUTDOWN BY USER
9DBCFDEE   0903101707 T O errdemon       ERROR LOGGING TURNED ON
192AC071   0903100407 T O errdemon       ERROR LOGGING TURNED OFF
A6DF45AA   0903100307 I O RMCdaemon      The daemon is started.
7F88E76D   0903100307 P S console        SOFTWARE PROGRAM ERROR
7F88E76D   0903100307 P S console        SOFTWARE PROGRAM ERROR
2BFA76F6   0903100107 T S SYSPROC        SYSTEM SHUTDOWN BY USER
9DBCFDEE   0903100207 T O errdemon       ERROR LOGGING TURNED ON
192AC071   0903095907 T O errdemon       ERROR LOGGING TURNED OFF
F3931284   0903095807 I H ent9          ETHERNET NETWORK RECOVERY MODE
EC0BCCD4   0903095807 T H ent9          ETHERNET DOWN
```

System Error Reporting – AIX II

```
AIX system error log example:gbsrual0048:root:/> errpt -a | pg
---------------------------------------------------------------------------
LABEL:          CONFIGRM_STARTED_ST
IDENTIFIER:     DE84C4DB
Date/Time:      Thu 13 Sep 16:09:22 2007
Sequence Number: 561
Machine Id:     0002A587D600
Node Id:        gbsrual0048
Class:          O
Type:           INFO
Resource Name:  ConfigRM

Description
IBM.ConfigRM daemon has started.

Probable Causes
The RSCT Configuration Manager daemon (IBM.ConfigRMd) has been started.

User Causes
The RSCT Configuration Manager daemon (IBM.ConfigRMd) has been started.

      Recommended Actions
      None

Detail Data
DETECTING MODULE
RSCT,IBM.ConfigRMd.C,1.29,207
ERROR ID
REFERENCE CODE
---------------------------------------------------------------------------
```

System Error Reporting Checkpoint I

1. Where are the system error log files kept for:

   a) Linux _____

   b) AIX _____

   c) Solaris _____

2. What command would you to display the system error logs for AIX?

   _____

System Error Reporting Checkpoint

3. Which command would you use to display the system error logs for the following flavours of UNIX:

   a) Linux _____

   b) AIX _____

   c) Solaris _____

4. What information is recorded in the system error logs?

_____

UNIX Fundamentals Part II

- vi editor
- Redirection & Regular Expressions
- Scheduling
- Networking
- Printing
- UNIX Tools & Utilities
- System Error Reporting
- **UNIX Manual Pages**

UNIX Manual Pages

- Most UNIX servers have help for UNIX commands in the form of manual pages, this includes Linux.
- The “man” command can be used to access this help.
- However, some production servers do not have the man pages installed in order to save space and for ‘security’ reasons.
- The man page database is grouped into eight sections:
  1. Indicates user commands and daemons.
  2. Indicates system calls and kernel services.
  3. Indicates subroutines.
  4. Indicates special files, device drivers, and hardware.
  5. Indicates configuration files.
  6. Indicates games.
  7. Indicates miscellaneous commands.
  8. Indicates administrative commands and daemons.
- All man pages follow a common layout that is optimized for presentation on a simple ASCII text display, possibly without any form of highlighting or font control.

UNIX Manual Pages I

- An example of a man page for the command “ls”.

```
$ man ls
------------------------------------------------------------------------------

Commands Reference, Volume 3

------------------------------------------------------------------------------

ls Command

Purpose

Displays the contents of a directory.

Syntax

To Display Contents of Directory or Name of File

ls [ -1 ] [ -A ] [ -C ] [ -F ] [ -L ] [ -N ] [ -R ] [ -a ] [ -b ] [ -c ] [ -d ]
[ -e ] [ -f ] [ -g ] [ -i ] [ -l ] [ -m ] [ -n ] [ -o ] [ -p ] [ -q ] [ -r ] [
-s ] [ -t ] [ -u ] [ -x ] [ File ... ]

To Display Contents of Directory
```

UNIX Manual Pages II

- If configured to do so, you can also use a keyword to find suitable command help.
- Examples:
  - "man -k rpcinfo", would give a list of commands where the word "rpcinfo" was found.
  - To display information about the grep command, enter: man grep
  - To display all matching entries, type the following:

    man -a <*Title*>
  - *To display only the first matching entry, type the following:*

    man -F <*Title*>

Manual Pages Checkpoint

1. How would you display inforamtion about the passwd command?

---

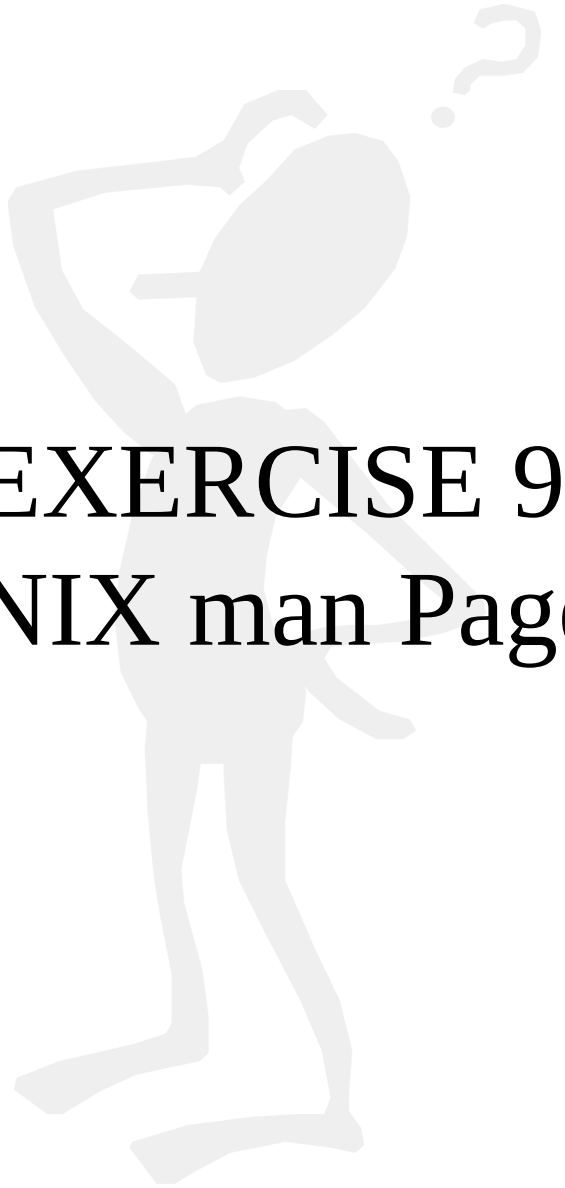
2. How would you display information about the rpc_$register library routine?

---

3. How would you display all matching entries for the command find?

---

4. How would you display the first matching entry in the man pages for the vmo command?



EXERCISE 9
UNIX man Pages

Fin

(End of Part II)

Thank you for attending!

Any Questions?