Document Number:	***Copy Number:***

TATA

TATA CONSULTANCY SERVICES

PRE ILP – Unix LOUNGE

Content Manual

Version 1.0

February 2014

TATA CONSULTANCY SERVICES

A Division of **Tata Sons Limited**

TATA

DOCUMENT RELEASE NOTICE

Notice No.
Client
Project
Document details

Name	Version No.	Author	Description

Revision details:

Action taken (add/del/chg)	Preceding page No.	New page No.	Revision description

Change Register serial numbers covered:
The documents or revised pages are subject to document control.
Please keep them up-to-date using the release notices from the distributor of the document.
These are confidential documents. Unauthorised access or copying is prohibited.

Approved by : ________________________________ Date: _________

Authorised by: ________________________________ Date: _________

DOCUMENT REVISION LIST

Client
Project
Document Name
Release Notice Reference (for release)

Rev. No.	Revision date	Revision description	Page No.	Prev page No.	Action taken	Addenda /New page	Release Notice reference

CONTENTS

CHAPTER 5 - UNIX FILE COMMANDS

5.1 Objective

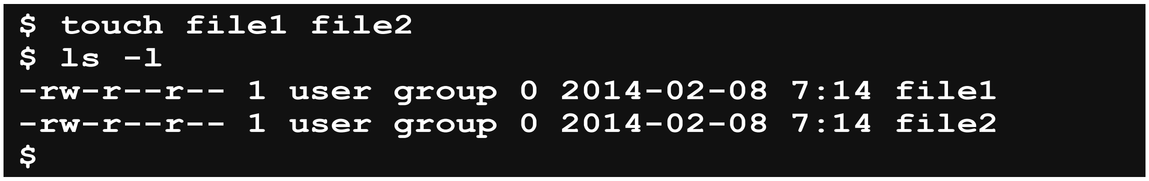
- To discuss the basic file commands.

5.2 Course Content

- touch
- cat
- cat > and cat >>
- find
- cp
- mv
- rm

5.2.1 touch command

Touch command is used in Unix to create an empty file.



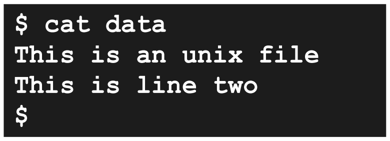
```
$ touch file1 file2
$ ls -l
-rw-r--r-- 1 user group 0 2014-02-08 7:14 file1
-rw-r--r-- 1 user group 0 2014-02-08 7:14 file2
$
```

Touch command is used to change these timestamps (access time and modification time of a file).

5.2.2 cat command

Use of cat command:
'cat' command is used to display a file content.

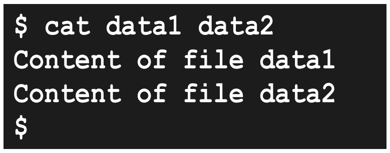
Syntax: $ cat filename [filename]



```
$ cat data
This is an unix file
This is line two
$
```

'cat' command can be used to display content of more than one file content. The name cat originated from the word concatenate

Syntax: $ cat file1 file2 ...



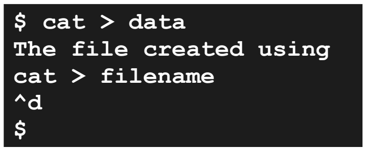
```
$ cat data1 data2
Content of file data1
Content of file data2
$
```

Note: cat command also changes the access time of the file.

5.2.3 cat > and cat >> command

Redirection operator (> and >>) can be used with cat command to take input from standard input device(keyboard) and store them to a file.

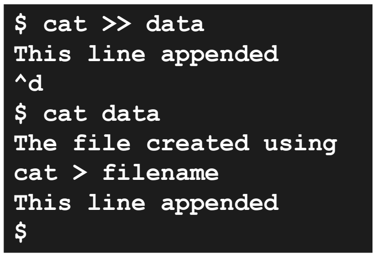
Syntax: $ cat > [filename]



```
$ cat > data
The file created using
cat > filename
^d
$
```

Press <ctrl>+d to save and exit after typing the contents of the file

cat >> command is used to append text to a file.
Syntax : $ cat >>[filename]



```
$ cat >> data
This line appended
^d
$ cat data
The file created using
cat > filename
This line appended
$
```

5.2.4 cp command

cp command is used to copy one file content to another file.

Syntax: $ cp [source filename] [destination filename]

Switches:
1. cp -i [sourcefilename] [destination filename]

This command copies the content of a file interactively.

2.cp -r [source_directory] [destination_directory]

This command copies the whole file hierarchy.
3. cp -v [sourcefilename] [destination filename]

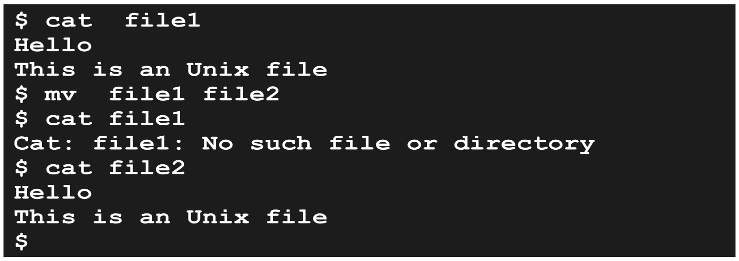
Copies in verbose mode, explains what is being done

5.2.5 mv command

mv command is used for:

1. for renaming a file in same directory.
2. for moving a file to a different directory.

Syntax:
    $ mv [sourcefilename] [destinationfilename]
    $ mv [source_directory] [destination_directory]



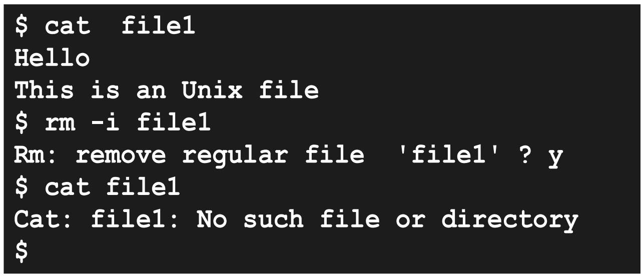
```
$ cat  file1
Hello
This is an Unix file
$ mv  file1 file2
$ cat file1
Cat: file1: No such file or directory
$ cat file2
Hello
This is an Unix file
$
```

5.2.6 rm command

rm command is used to delete file.
Syntax: $ rm [filename]

Switches:
1. $ rm -i [filename]
This command is used to remove a file interactively.

2. $ rm -r [directory_name]
This command is used for delete the content of directory recursively



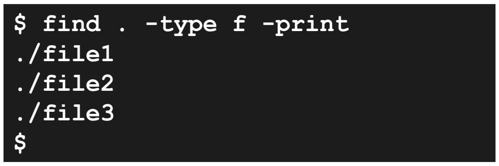
```
$ cat  file1
Hello
This is an Unix file
$ rm -i file1
Rm: remove regular file  'file1' ? y
$ cat file1
Cat: file1: No such file or directory
$
```

5.2.7 find command

- find command is used to locate the existence of a particular file in unix.
- With find command one can search file based on name, type,group and other attributes of file.
- find command is used to locate unused file also.

Syntax: find <path> <selection_criteria> <action>
$ find . -type f -print

Finds only the regular files from the current directory.



```
$ find . -type f -print
./file1
./file2
./file3
$
```

Examples:

Command	Explanation
find . -name “*.c” -print	Finds files with '.c' extensions from current directory & prints.
find / -name '[A-Z]*' -print	Finds filles starting with capital A to Z from root directory.
find . -type d -print	Finds all the directories from current directory and prints it.
find / -mtime -2 -print	Find all files modified two days ago from root.
find . -atime -2 -print	Find all files accessed 2 days ago from current directory.
find / -name a.out -print	Find file named a.out from root.

5.3 Video 5: File Commands

http://www.youtube.com/watch?v=BM2D6uNpMoU&list=PL8A83A276F0D85E70

5.4 Quiz Time

Q1. Which command is used to recursively copy the files?
A. cp -v
B. cp -a
C. cp -r
D. cp -all

Answer: C

Q2. Which command can be used to append some text at the end of some file?
A. append <filename>
B. cat <filename>
C. cat >> <filename>
D. cat > <filename>

Answer: C

Q3. A file's modification time can be changed by using;
A. change
B. newaccess
C. touch
D. cat

Answer: C

Q4. rm is used to ______ a file:
A. remove
B. rename
C. remodel
D. rectify
Answer:A

Q5. find command is used to:
A. locate a particular file
B. locate all similar files
C. locate all directories
D. all of these
E. none of these
Answer: D

CHAPTER 6 - BASICS OF VI EDITOR AND ENVIRONMENT VARIABLES

6.1 Objective

- To understand the use of vi editor and the concept of Environment Variables.

6.2 Course Content

6.2.1 The vi Editor

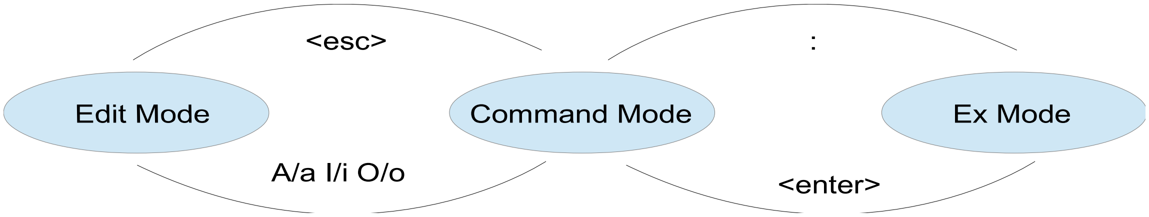
It is a visual editor used to enter and edit text files. Invoking vi with/without filename puts it in command mode:

```
vi [<file name>]
```

vi works in three different modes:

- Edit Mode - where any key is entered as text
- Command Mode - where keys are used as commands
- Ex Mode - ex commands can be entered in last line to act on text

Relationship between 3 modes



6.2.1.1 Edit Mode Commands

Command	Action
a	Appends text to right of cursor
A	Appends text at end of line
i	Inserts text to left of cursor
I	Inserts text at beginning of line
o	Inserts blank line below + inserts text
O	Inserts blank line above + inserts text

rx	Replace current character with char x
Rtext<Esc>	replaces character with text

6.2.1.2 Save and Exit from vi

Save and Exit Commands of ex Mode

Command	Action
:w	save file and remain in edit mode
:wq	save file and quit edit mode
:w <filename>	write buffer to file name
:q	quit editing mode when no changes are made
:q!	quit editing mode, abandoning changes
:x	save file and quit editing mode

6.2.1.3 Cursor Movement Commands

Command	Action
h or backspace	Moves Cursor Left
l or spacebar	Moves Cursor Right
j	Moves Cursor Down
k	Moves Cursor Up
nG	Go to line number n
$	Moves cursor to end of line
w	Moves cursor to next word
b	Moves cursor back to previous word
e	Moves cursor forward to end of word

Note: W, B, E perform functions similar to w, b, e except that punctuation is skipped)

6.2.1.4 Search and Repeat-Search Commands

Command	Action
/pat	Search forward for pattern pat

?pat	Search backward for pattern pat
n	Repeat search in previous search direction
; or ,	Repeat last character search in same or opposite direction (by f/F/t/T) only in current line (repeat factor works)

6.2.1.5 Deletion and Modification Commands

Command	Action
dw	Delete Current Word
dd	Delete Current Line
d$ or D	Delete from cursor to end of line
cw	Change Current Word
cc	Change Current Line
c$ or C	Change from cursor to end of line
x	Delete character in current cursor position
X	Delete character before cursor
.(dot)	Repeat last editing instruction
u	Undo single last change
U	Restore all changes to line since cursor moved to it

6.2.1.6 Commands to Move or Copy Lines

[“<named-buffer>][n]dd
Cut (delete) n lines starting from current line

[“<named-buffer>][n]yy
Copy n lines starting from current line

[“<named-buffer>]p
Put yanked text after current cursor position

[“<named-buffer>]P
Put yanked text before current cursor position

Note: Named-buffer is useful for copying an area from one file into another. Open some files simultaneously by vi. Mark an area in one file by dd or yy; move to another file (by :next) to paste (by p or P) the named area; then say :rewind to return to the parent file.

6.2.1.7 Customizing vi editor environment

vi can be customized as per users' requirements using ex-mode commands.

Ex Commands	Action
:set all	Display all set options; those pre-fixed with no are not operative
:set autoindent(ai)	Extremely useful to programmers for indentation of lines
:set number (nu)	Display all lines with numbers which are not preserved on saving file
:set nonumber(nonu)	Reverses number setting
:set tabstop=6 (ts)	Changes default tab setting (8 spaces)
:set ignorecase(ic)	ignores case while pattern matching
:set showmatch (sm)	Helps locate matching brackets when) or } is entered in input mode; beeps when no match found to alert for correction
:set autowrite (aw)	Writes current file automatically whenever switching files with :n and escaping to shell with :sh
:next (n for short)	Moves to next file opened in vi
:rewind	Comes back to parent file

All sets can be stored in .exrc file used by vi for its startup instructions.

Equivalently an environment variable, EXINIT can be assigned to store the settings:

$EXINIT="set nu tabstop=6 ignorecase"

Assignment of these changes in the .profile will be available for all sessions.

6.2.2 Environment Variables (EVs)

Environment variables are the variables that control the behaviour of the system. Every UNIX process runs in a specific environment . An environment consists of a table of environment variables, each with an assigned value. When a user logs in, some files are executed and they initialize the table holding the environment variables for the process. This table is accessible to the shell. A child process inherit the entire execution environment from the parent. Environment variable names are generally given in upper case by convention.

To get a full list of currently set environment variables, 'env' command is used.

Some examples of the environment variables are the USER, LOGNAME, HOME, PATH, PS1, PS2, TERM MAIL, etc.

- **The HOME Variable** : It specifies an associated directory with every user in a UNIX system. If the HOME variable for the user Sita contains /usr/sita/stores, every time Sita logs in, she is taken to the directory stores.

The variable HOME is referenced the same way: $ echo ${HOME}<Enter>

- **The PATH Variable** : Contains a list of all full path-names (separated by a colon) of directories that are to be searched for an executable program. For example, the command $PATH=.:/usr/bin:/bin<Enter> specifies directories to be searched for any executable file or a command file (current directory, /usr/bin and /bin, in that order).
- **The PS1 Variable :** The system prompt may be changed by setting the value of this variable to the desired prompt:
  $ PS1="Hello>"<Enter>
- **The PS2 Variable**
  prompt string for continued command line (default '> ').
- The PWD Variable
  The current working directory can be displayed:
  echo "${PWD}"
- **$IFS**: String of characters which are used as word separators in command line (space, tab, newline chars).
- The **MAIL Variable :** Names the standard file where your mail is kept

The .profile File

This is a startup script which is executed when the user logs in. By setting values to environment variables, user can change the working environment for the session. However to make the environment setting permanent , the values to the environment variables needs to be set in the .profile file. The .profile must be located in the home directory of the user. There is an universal environment setting also, maintained by the administrator in /etc/profile available to all users.

A sample .profile file would look like:

```
PATH=.:/bin:/usr/bin
 export HOME PATH PS1 MAIL
```

6.3 Video 6: Working with vi editor

http://www.youtube.com/watch?v=njlQiSAbSA0

6.4 Quiz Time

Q1. Which of the following is not a navigation key in vi editor?

A. h
B. k
C. l
D. g

Answer: D

Q2. vi editor does not work in which mode?

A. edit
B. command
C. ex mode
D. read only

Answer: D

Q3. Multiple lines can be deleted in vi using,

A. dd
B. ndd, where n is the number of lines to be deleted
C. rm
☚蹊 nrm, where n is the number of lines to be deleted

Answer: B

Q4. Which of the following is not a Environment Variable?

A. PATH
B. NAME
C. PS2
D. HOME

Answer: B

Q5. To customize the vi editor, the action should be prefixed with:

A. :prefix
B. :set
C. :nonull
D. :custom

Answer: B