**Part I – Unix Programming**

**1) Exercise I: Basic unix commands**

|  |  |  |
| --- | --- | --- |
| **Command** | **How to use** | **Meaning** |
| login |  | Login to system |
| passwd | >passwd | To change the passwd of your login |
| tput clear | >tput clear | To clear the screen |
| pwd | >pwd | To display the present working directory |
| ls | >ls  >ls chap\* | To list all the files in you directory  To list all the files whose name starting with ‘chap’ |
| cat | >cat filename | To see the content of the file ‘filename’ |
| cd | >cd new | To change the current directory to new |
| date | >date | To display the system date |
| who | >who | To list the users who are presently using the system |
| Redirection  operator | >who > a | The output of ‘who’ command is stored in the file ‘a’ |
| wc | >wc a | To count the lines, words and characters in the file ‘a’ |
| Pipe | >who | wc | To connect two commands.  The output of who command is transferred as input to the wc command. |
| echo | > x= 5  > echo $5  5  > echo hello  hello  >echo one two three  one two three | To display the value in a variable.  To display the hello message.  To display more than one word. |
| Exit | >exit | Terminates the session. |
|  |  |  |
| Try  > >a.txt  > try pipe between different commands | | |

**2) Exercise II: Unix File system**

|  |  |  |
| --- | --- | --- |
| **Command** | **How to use** | **Meaning** |
| ls | >ls -x | To produce multicolumn output.  (you can see the difference if your directory has around 100 files and subdirectories) |
|  | >ls -F -x | \* is added to exe files, / is added for directory files. |
|  | >ls -l | To display long list, lists full information about each of the file. |
|  | >ls -l a b | To display information only about the files a and b. |
|  | >ls –l dir1, dir2 | To display long list of files in the subdirectories dir1 and dir2. |
|  | >ls -l -d dir1, dir2 | To display the attributes of the subdirectories dir1 and dir2, not the files inside them. |
|  | >ls -r | To display the list in reverse order. |
|  | >ls -a | To display all the files including hidden files. |
|  | >ls -t -l | To display last modified time of files. |
|  | >ls -R | To display all the files, directories and files inside the directories recursively. |
|  | >ls -R -r -a | To display all the files in reverse order, directory wise. |
|  | >ls -u | To sort the files by access time. |
|  | >ls -i | To show the i-node number of the file. |
|  | >ls -s | To display number of blocks used by a file. |
|  | >ls -1 | To display one file per line. |
| cat | >cat > a | To create file in input mode. |
| mkdir | >mkdir nd1  >mkdir nd2 nd2/sd1 nd2/sd2 | To create a subdirectory nd1.  To create a subdirectory nd2, with subdirectories sd1 and sd2. |
| rmdir | >rmdir nd1  >rmdir nd2/sd1 nd2/sd2 | To remove the subdirectory nd1  To remove the subdirectories sd1 and sd2 of the directory nd2.  (to delete a directory, the directory should be empty). |
| cp | >cp file1 file2  >cp file1 nd/file2  >cp file1 file2 file3 nd | To copy content of file1 to file2.  To copy content of file1 to file2 in the subdirectory nd.  To copy file1, file2 and file3 to the subdirectory nd. |
| rm | >rm a  >rm a b c  >rm nd/a  >rm \*  >rm -i a  >rm -r \*  >rm -r \* -i | To remove a file.  To remove the files a, b and c  To remove the file ‘a’ from the subdirectory ‘nd’  To remove all the files from the present directory.  To remove interactively.  To remove all the files in the directory.  To remove all the files and directory interactively. |
| mv | >mv file1 file2  >mv file1 dir1  >mv file1 file2 file3 dir1  >mv dir1 dir2 | To rename the file file1 and file2.  To move file1 from current directory to the directory dir1.  To move more than one file.  To rename the directory. |
| chmod  u-user  g-group  o-others  r-4, w-2, x -1  rw-6, wx-3, rx-5  rwx-7 | >chmod u+x f1  >chmod ugo+x f1  (or)  >chmod a+x f1  >chmod g+r f1 f2 f3  >chmod go-r f1  >chmod u-x, go+r f1  >chmod o+wx f1  **(to assign absolute permission)**  >chmod ugo=r file1  (or)  >chmod a=r file1  (or)  >chmod =r file1  **(octal notation)**  >chmod 635 file1  >chmod 644 file1  >chmod 777 file1  >chmod 000 file1  >chmod 6 file1  >chmod 56 file1  >chmod 456 | To add executable permission to user for file 1.  To add executable permission to all for file 1.  To add read permission to group for the files f1, f2 and f3.  To remove read permission from group and others for the file f1.  To remove and add permission at the same time.  To add write and executable permission to others for the file f1.  To add only ‘read’ permission to all.  To set rw for user, wx for group and rx for others.  To set rw for user and r for group and others.  To set rwx for user, group and others.  To remove all the permission from all.  To set rw for others.  To set rw for others and rx for group.  To set r for users, rx for group and rw for others. |
| General purpose utilities  1) Halted output  2) File types  3) Word count  4) Octal Dump  5) Comparing two files.  6) Further comparison | >more f1  >more f1 f2 f3  > bc  >file f1  >file dir/\*  >wc f1  > wc –l f1  >wc –w f1  >wc –c f1  > wc –wc f1  >od a  >od –b a  >cmp f1 f2  >cmp –l f1 f2  >comm. f1 f2  >comm. -3 f1 f2  >comm. | (more command is used as paging tool, to view single page at a time)  It has certain internal commands  :n – switch to next file  :p – switch to previous file  :f – to get file name  z – scroll forward.  s – skip one line.  d – scroll forward.  f- skip all the line  ‘ (single quote) – go back.  V – enter to VI, edit and come back.  To enter into calculator.  ^d to come out.  (to determine type of file)  To see file type of all the files in the directory dir.  To count the line, word and character.  To count only lines.  To count only words.  To count only characters.  To count word and characters.  To display octal value of the file content, each character is replaced by its octal value.  Each line displays 16 bytes of data.  To files are compared and the location of first mismatch is returned.  f1 f2 differ : char 9, line 1  To get detailed list of byte number and the differing bytes in octal for each character differ in both the files.  Both files should be in sorted order.  It gives three column output “lines unique to f1”, “lines unique to f2” and “lines common to f1 and f2”.  Only column 1 is displayed. |
|  | >comm. -12 f1 f2 | Only column 3 is displayed. |
| 7) File difference with diff  8) Printing a file  9) others  10) Login details. | >diff f1 f2  >lp f1  >lp –t “title” f1  >banner UNIX  >cal 1991  >cal 01 2004  >date +%m  >date +%h  >date +%d  >date+%d  >who  >who –H  >who –Hu  >whoami  >who am i  >tty  >stty  >sty -a | To display which lies in one file have to be changed to make the two files identical.  The file f1 will be submitted and prompt is returned immediately.  To print month.  To print month name.  To print the day of the month.  To print two digits of the year.  To display user name, terminal, date and time of logging in.  To display all columns with header.  To show the idle status  . – not idle.  00:04 – idle for 4 minutes.  To display the user name.  To display all other details.  To know your terminal.  To set the parameter to our terminal.  To display all current setting. |

**Exercise III: Simple Filters**

|  |  |  |
| --- | --- | --- |
| **Simple Filters**  **1)** To create a simple database  2) Paginating files.  3) Displaying the beginning of a file. | >cat > emp.lst  No | Name | Post | Salary  12 | Raj | Secretary | 2000  ;  ^d  >pr emp.lst  >pr emp.lst > out.lst  >pr emp.lst | lp  >pr < emp.lst  >pr –l72 emp.lst  >pr –w132 emp.lst  > pr –h “code list” emp.lst  >pr –n –h “code list” emp.lst  >pr -020 dept.lst  >pr -3 emp.lst  >pr –m emp1.lst emp2.lst  (or)  >cat emp2.lst | pr –m emp1.lst  >pr –m –s “\*” emp1.lst emp2.lst  >pr +2 emp.lst  >pr –d emp.lst  >pr –l80 –w80 –n –d –m emp1.lst emp2.lst  >pr –t emp.lst  >head emp.lst  >head -3 emp.lst  >head -1 emp| wc –c  (or) | (commands which are doing some filtering action on the data input is called filters)  To create simple data base.  To prepare files for printing with headers.  To change lines per page: 72.  To change width per line to 132.  To display the header.  To display the line number.  To shift 20 spaces to right.  To display 3 column output.  To merge two files.  To add column separator.  To start printing from 2nd page.  To add blank line for every line.  To omit header and footer.  To display the first 10 records.  To display first 3 records.  To find the record length. |
| 4) Displaying the end of a file  5) Splitting a  file vertically  6)Pasting files | >reclen=’head –l emp |wc –c”  > echo $reclen  $head -2 emp1 emp2  >tail emp  >tail -3 emp  >tail +11 emp  >tail -10c emp  >tail -1b emp  >tail –f emp  >head -5 emp.lst>shortlist  >cat shortlist  >cut –c6-22, 24-32 shortlist  >cut –c-3 emp.lst  >cut –c5- emp.lst  >cut –d”|” –f2,3 emp.lst  >cut -d”|” –f1,3- emp.lst  >paste file1 file2  >paste –d”|” file1 file2  >paste –d”|#~” file1 file2 file3 file4  >cut –d”|” –f1,4- shortlist | paste –d “|” cutlist-  (or)  >cut –d”|” –f1,4- shortlist | paste –d “|” -cutlist | To display first two records of both the files.  To display the last 10 lines.  To display the last 3 lines.  To display all records from 11th record.  To display last 10 characters.  To display the last one block (512 bytes).  To see the growth of the file from another terminal.  (head and tail commands split the file horizontally)  To cut vertically from characters 6 to 22 and from 24 to 32.  To cut first three characters of all the records.  To cut from 5th character to all characters.  To cut fields 2 and 3. –d to specify delimiter and –f to specify the fields.  To cut fields 1, 3 and others.  (‘cat’ pastes files horizontally and ‘paste’ pastes files vertically)  To paste file1 an d file2 vertically.  To paste file1 and file2 vertically, with delimiter. |
| 7) Ordering a file  Sorting on secondary key.  Sorting on column.  Numeric sort.  uniq command | >sort cutlist1  >sort –t”|” +1 emp.lst  (or)  >sort +1 emp.lst  >sort –r +1 emp.lst  Or  >sort +1r emp.lst  >sort –o newfile +1 emp.lst  >sort –o emp.lst emp.lst  >sort –c file  >sort –t”|” +3c file  >sort –t”|” +2 +1 emp.lst  (or)  >sort +2 +1 emp.lst  >sort –t”|” +1.2 -1.5 emp.lst  >sort –m file1 file2 file3  >cat Numfile  2  3  4  10  27  ^d  >sort Numfile  >sort –n Numfile  >sort –u –n Numfile  >cut –d”|” –f3 emp.lst | sort –u | tee x.lst  >uniq emp.lst  >sort emp.lst | uniq  >sort emp.lst | uniq –t | To sort the content of a file.  To sort on specified field, +1 means sort on second field.  To sort in reverse order by 2nd field.  To store the sorted output in new file.  (sort is the only command which has both input and output files the same)  To check whether the file is sorted.  To check whether the file sis sorted on fourth column.  **To sort on secondary key.**  The file will be sorted on filed 2 and then by field 3.  **To specify the character position.**  To sort by 2nd field’s 3rd and 4th column.  To merge the sorted output of file1, file2 and file3.  Error.  To sort the numbers in the file.  To remove the duplicates (-u – unique)  To select unique designation from emp.lst.  To display all but only one copy of redundant record.  The input to the uniq can be sorted.  Writes to t. |
| Line numbering | >sort emp.lst | uniq –u  >cut –d”|” –f3 emp | sort |uniq –u.  >sort emp.lst | uniq –d  >sort emp | uniq –c  >nl filename  >nl –w2 –s”|” emp  >nl –w2 –s”|” –nrz emp  >nl –w2 –s”|” lz emp  >nl –v10 emp  >nl –v10 –i2 emp | To display only non repeated records.  To display non-repeated values in the 3rd column of emp.  To display the redundant records, but only once.  To display frequency of occurrence of repeated lines.  To number the lines in the file.  To use delimiter between the number and the first field. (-w2 width for number).  To right justify with leading zero.  To left justify with no leading zero.  Numbering will start from 10.  Numbering will start from 10 and incremented by 2. |

**Exercise 4: Advance Filters**

|  |  |  |  |
| --- | --- | --- | --- |
| Usage of single quotes and double quotes  Searching for a pattern. | >echo ‘Hello’  >x = 5  >echo “$x”  >echo “\date\”  >echo “\$x”  Output : x  >grep sales emp  >grep sales emp1 emp2  >cat emp?.lst | grep sales  (or)  >grep –h sales emp?.lst  >grep sales man emp  >grep ‘sales man’ emp  >str=’sales man’  >grep “$str” emp.lst  >grep –c ‘sales’ emp?.lst  >grep –n ‘sales’ emp  >grep –v ‘sales’ emp | tee otherlist  >grep –l ‘sales’ emp?.lst  >grep –i ‘SALES’ emp | Use single quote to enclose a command argument.  To display the value of x.  A pair of back quotes helps for command substitution.  A single back slash escapes the special character.  grep – global regular expression printer.  To display all lines with string “sales”.  To use two files.  grep can not open man, records with sales in emp will be displayed.  To count the occurrence of sales in each file.  Output:  emp1.lst : 2  emp2.lst : 3  To display the line number which containing ‘sales’.  To select all the lines which are not having the string ‘sales’, display the output and store the result in a file called ‘otherlist’.  To display only the file names which consists of string ‘sales’.  To ignore the case of string. | |
| grep with regular expressions | ch\* - matches to 0 or more occurrence of ch.  [pqr] – matches to single char p, q or r.  [c1-c2] – range.  [^pqr] – matches to characters which are not p, q or r.  ^ptn – matches at the beginning.  ptn$ - matches at the end.  ch\{m\} – matches m number of occurrence of ch.  ch\{m,b\} – matches m to n occurrence of ch.  ch\{m,\} – matches to minimum of m occurrence. | | |
|  | >grep ‘[cC]ho[wu]dh\*ury  Output:  choudhury  Chowdury  >grep ‘[7..9]…$’ emp  >grep ‘ | | [cC] – c or C  [wu] – w or u  h\* - 0 or more hs  To select lines whose salary is 7000 or more. |