CSE2031

Lab 2
FALL 2009

In this lab, you will be introduced to more complex Unix commands. After this lab, you should be comfortable using Unix/Linux in the lab and as a platform for software development.

File names generation

When you enter file name for a command (cat filename) there is a way of generating a list of files according to a specific pattern. Some characters or patterns have a special meaning to the shell, we can use them to generate this list, these characters/patterns are [much more on this later]

char/pattern	Description
*	Matches any string of characters including th null string but not a leading dot
?	Matches any single character
[abcf]	Matches any one of the characters in the list
[a-x]	Matches any characters between a to x (inclusive)
[^abc]	Matches any character other than a,b,c (matches any character not in the set

So for example

```
ls   test*
```

Lists files that starts with test (test1, test.txt, test.doc, test_1, testfinal, …. including a file names test if they exist.)

```
ls test?1
```

Lists files such as test_1 testa1, test21, but not test1

ls test[ab].txt

Lists testa.txt, testb.txt but not textc.txt

ls test[a-d].txt

Lists testa.txt, testb.txt, testc.txt, and testd.txt if they exist in the3 current directory.
Try echo * what does it do?
If a special character (such as *) is quoted using \ as * looses its special meaning and acts like any other characters, for example

ls test* lists only the file "test*" if it exists. Also \ is used to escape end of line character so it does not mean the end of the command, for example

```
tigger 193 % echo hi this is\
? a multi line echoing
hi this is a multi line echoing
tigger 194 %
```

sort

The command sort sorts the input file(s) according to some key. The input stream is treated as independent records of variable length delimited by new lines. Each record is treated as fields delimited by whitespaces or user specified single character. For a complete list of options, man sort. Some of the options are mentioned here in the examples below.
The two most popular options are -t and -k
-t specifies the character that delimits fields, for example -t: specifies that field are terminated by the character ":"white spaces are part of the field. Without specifying -t, fields are separated by white spaces and leading and trailing white spaces are ignored. Note the difference between using -t' ' and not specifying -t at all. In the first case, a record with space followed by A followed by space is a record with three fields, while not specifying -t interprets it as a single filed record (A).
The other popular option is -k where the field number is specified, if you use -k4 that means sort based on the 4th field.
Consider the file file1

```
sh-3.00$ cat file1
this is line 1
zsomething starts with 2
this is line 2
something starts with 2
this is happy line
this is line 21
this is line 200
this is line -5
line without numbers
sh-3.00$
```

Noe if we use the default sort.
The output is as follows

```
sh-3.00$ sort file1
line without numbers
something starts with 2
this is happy line
this is line -5
this is line 1
this is line 2
this is line 200
this is line 21
zsomething starts with 2
sh-3.00$
```

here the sorting based on filed 1, if tie use field 2 and so on.

```
sh-3.00$ sort -k4 file1
line without numbers
this is line -5
this is line 1
something starts with 2
this is line 2
zsomething starts with 2
this is line 200
this is line 21
this is happy line
sh-3.00$
```

Here the sorting based on the fourth field as a key the first line has no 4^{th} field, i.e. empty key that comes before anything. Note that 200 comes before 21 since 200 is treated like a string, where 0 comes before 1. If we want to treat the field as integer, use the -n flag

```
sh-3.00$ sort -k4 -n file1
this is line -5
line without numbers
this is happy line
this is line 1
something starts with 2
this is line 2
zsomething starts with 2
this is line 21
this is line 200
sh-3.00$
```

The format -kn means the key starts at the n^{th} field and continue to the end of the record. If the format is -kn,m the key starts at the beginning of the n^{th} field and ends at the end of the m^{th} field. If the format is -k2.3,4.1 it means the ket starts at the 3^{rd} character of the second field and ends at the first character of the

4th field.

uniq

Uniq removes duplicate records from data stream. Usually it is used with sort to sort the file then remove duplicate records. Sort -u removes records with duplicate keys.
Example: sort file |uniq
Sorts the named file, then the output is pipelined to uniq to remove duplicate records.

cut

The cut command is used to extract a portion of the file. It can extract based on either fields or character positions. For example
cut -c1-3 file
Displays character 1,2, and 3 from each record.
While cut -f1-3 file
Displays fields 1,2, and 3 from each record

The default delimiter is a tab, but you can change it with the -d option.

tr

The command tr translates characters from in a file.

Tr [options] string1 [string2]
Characters in string1 is translated into characters in string 2 character by character.

join

The join command merge records in **sorted** files based on a common key
for example
if we have two files, suers and susers1

```
sh-3.00$ cat suers
cat: suers: No such file or directory
sh-3.00$ cat susers
jdoe    John Doe        4/15/96
jhsu    jack Hsu        1/2/93
lsmith  laura Smith     3/12/96
pchen   paul Chen       1/5/97
```

```
que     extra field
sphilip Sue Philip      4/4/94
sh-3.00$ cat suers1
cat: suers1: No such file or directory
sh-3.00$ join -1 1 -2 1 susers susers1
jdoe John Doe 4/15/96 John Doe 4/15/96 External
jhsu jack Hsu 1/2/93 jack Hsu 1/2/93 Internal
lsmith laura Smith 3/12/96 laura Smith 3/12/96 Internal
pchen paul Chen 1/5/97 paul Chen 1/5/97 EX
sphilip Sue Philip 4/4/94 Sue Philip 4/4/94 Internal
sh-3.00$
```

where join -1 1 -2 1 susers suers1 means join the two names files based on the first field of the first file (-1 1) ans the first field of the second file (-2 1)
For complete details, look up man joint

Exercise
Do the following operations using the least number of commands (use pipelining and redirection to minimize the number of Linux commands).

1. Display the permissions field of the files in your current directory (for example -rwx—x--x).
2. Display the count of c files in your current directory (assume that only c files ends with *.c)
3. Display the month each file in the current directory was last modified (check the -s option of tr)

Consider the file /etc/passed. This file contains information about users in the system. The file consists of records, each record is in a separate line, each record consists of (taken from `man 5 passwd`).

There is one entry per line, and each line has the format:

account:password:UID:GID:GECOS:directory:shell

The field descriptions are:

account	the name of the user on the system. It should not contain capital letters.
Password	the encrypted user password or a star.
UID	the numerical user ID.
GID	the numerical primary group ID for this user.
GECOS	This field is optional and only used for informational purposes. Usually, it

contains the full user name. GECOS means General Electric Comprehensive Operating System, which has been renamed to GCOS when GE's large systems division was sold to Honeywell. Dennis Ritchie has reported: "Sometimes we sent printer output or batch jobs to the GCOS machine. The gcos field in the password file was a place to stash the information for the $IDENTcard. Not elegant."

directory the user's $HOME directory.

Shell the program to run at login (if empty, use /bin/sh). If set to a non-existing executable, the user will be unable to login through login(1).

4. Display on the monitor for every user name and UID separated by a colon. Every user is on a separate line
5. As above but display UserID, and login shell separated by a tab
6. Display users names whose login shell is /bin/false

Consider the file ~/moktar/Public/busers. This file contains records with 2 fields each, login name and phone number.

7. Join this file with the /etc/passwd file to get a file with 3 fields per record, login, UID, and phone number

Submit a single text file with the answer top the above 7 questions to l1 (the letter small L and the number 1)