

Unix Tools  
Courant Institute of Mathematical Sciences  
Midterm Exam  
March 11, 2008

Give a concise answer for each of the following questions.

1. The command `last` lists the users who logged on the system and other information as follows.

```
i5$ last | head -3
mxx2876 pts/11      207-237-206-5.c3 Sun Mar 19 23:11 - 23:17 (00:05)
mxx2876 sshd       207-237-206-5.c3 Sun Mar 19 23:11 - 23:12 (00:00)
troop75 pts/10      pool-70-107-167- Sun Mar 19 23:10 - 23:28 (00:18)
```

Give a simple pipe using `last`, `cut`, `sort`, and `uniq` to list the users who logged on and the numbers of times each logged on, in decreasing order.

```
last | cut -d' ' -f1 | sort | uniq -c | sort -nr
```

2. Use `egrep` to search for all lines of file `foo.txt` containing a sequence of letters of length four or more starting with the same two characters it is ending with.

```
egrep '([a-z])([a-z])[a-z]*\1\2' < foo.txt
```

3. Each line file `foo.txt` is a sequence of digits. Use `grep` to print the lines of file `foo.txt` with at most 3 digits or at least 7 digits.

```
grep -v '^ [0-9]\{4,6\}$' foo.txt
```

4. Use `sort` to sort a tab-separated input file `foo.txt` according to the third column in reverse numerical order.

```
sort -t$'\t' -nrk3 foo.txt
```

5. File `foo.c` contains function calls to `f` with at least four arguments, e.g., `f(x,y,z,t)` or `f(x,y,z,t,u)`. Use `sed` to permute the first and last argument of `f` only when it is called with four arguments, e.g., `f(1,2*3,4+5,6)` becomes `f(6,2*3,4+5,1)`.

```
sed 's/f(\([^,]*\),\([^,]*\),\([^,]*\),\([^,]*\))/\
f(\4,\2,\3,\1)/g' foo.c
```

6. Use `find` to find all the files in the current directory that have been modified in the last twenty four hours or that have permission 666 and to make them unwritable to others.

```
find . \( -perm 666 -o -mtime 0 \) -print -exec chmod o-w '{}' \;
```

7. Write a `while` loop in `bash` that reads from `stdin` an input sequence and prints that sequence, and finishes if the length of that sequence is equal to the length of your last name.

```
while [ ${#name} -ne 5 ]
do read name
echo $name
done
```

8. Write a `bash` script that takes as arguments two strings  $x$  and  $y$  and changes filenames in the current directory by replacing a prefix  $x$  with the new prefix  $y$ , e.g., `foo1f.html` becomes `bar1f.html` if  $x = \text{foo1}$  and  $y = \text{bar1}$  (*hint*: you can use the parameter expansion functionalities available under `bash` and `ksh` such as `${#pattern}`).

```
#!/bin/bash

old=$1
new=$2

for f in `ls $old*`
do
    mv $f ${f/#$old/$new}
done
```

9. Write a simple `ksh` script that sends by email to each user in `/etc/passwd` that is not yourself all of its arguments and sign it with your own first name and last name.

```
#!/bin/ksh

for uu in `BEGIN{ FS = ":"}NF > 1{ print $1}` < /etc/passwd `
do if [ $uu != $USER ]
then mail $uu << EOM
$*
M. Mohri
EOM
done
```

10. The current directory contains a very large number of files. Write an awk script that prints the number of files owned by each user in this directory as well as the total number of bytes.

```
#!/bin/awk -f

BEGIN{ nfiles[""] = 0 }
{ nfiles[$3]++; tspace[$3]+=$5; }
END{
    for (x in nfiles) {
    if (x != "") print x, nfiles[x], tspace[x]
    }
}
```