LINUX commands and scripting

A breadth of knowledge in the use of LINUX commands can significantly improve the efficiency and ease with which a user can interact with the complex and voluminous data associated with the use of sophisticated computer packages such as AIMPRO. Here, some tasks are set test your knowledge.

1. The following are tasks often undertaken. In each case indicate the LINUX command or commands that are most likely to achieve the aims <u>efficiently</u>:
    a. Find all files created in the past 30 days.
    b. Find all files modified in the past 30 days.
    c. Copy all files matching a specific file name from one computer to another.
    d. Determine which out of a set of sub-directories contains the largest cumulative file size.
    e. List all text-files that contain the word 'optimise' exactly twice within a directory structure.

2. In each of the following commands, use the 'man' command to determine the operation of the command, the meaning of the associated command-line flag(s), and any values that are associated with them.

Command	Flag(s)
ls	-A, -F, -a, -l, -t
find	-name, -not, -mtime, -exec
cp	-R, -i
ssh	-X
du	-s, -h, -k
df	-H, -h
kill	-9
grep	-i, -c
mkdir	-p
rm	-r
chmod	-R

3. Under normal circumstances, one is expected to be running the t-shell (/bin/tcsh) on Snufkin, Verity and Trueman. In order to improve the efficiency of performing repetitive procedures, it is often useful to write short shell-scripts. For the following example, explain what each step does, and summarise the operation of the script as a whole.

```
#!/bin/tcsh –f
foreach d ( aim*.sh.o*)
  gres –Rlast $d
  mkdir rerun_$d
  mv dat.$d rerun_$d
  cd rerun_$d
  ln –s ../hgh-pots .
  ln –s dat.* dat
  ompisub 1x8 ~/bin/AIM
  cd ../
end
```

4. Write a shell-script to find and compress all files in a specific users file-system that are not currently compressed, omitting the home directory, ~/bin and ~/lib.
5. The use of 'aliases' for commands frequently employed can be very effective. In each of the following, determine whether the alias is well constructed, and determine either what the alias does, or (in the case(s) the alias is poorly written) what the most likely intended outcome might be:
    a. alias dir "echo */"
    b. alias test ~/bin/aimpro
    c. alias a alias
    d. alias true "ssh –X trueman.ncl.ac.uk"
    e. alais ll "ls –l"
    f. alias gx "gres –x"
    g. aimdmp "aimview --d=\!*"
6. Design an alias to perform the task defined in Q4.
7. An alternative to defining an alias is writing a script that is made executable and exists in the path specified for the machine.
    a. How is the 'path' defined in tcsh?
    b. How does the order of the directories in the path affect the way commands operate?
    c. What might influence the decision as to whether a command is written as a script accessible through the path or is defined as an alias?