Software Tools

C, Unix (Linux), and tools

Wild cards and Regular Expressions

- Any string: *
- Any single character: ?
- Any of a set of characters: [A-G0-9]

File Permissions

- Unix/Linux permisions recognize three kinds of people: the user (or owner of the file), the group, the others.
- And three kinds of permissions: read, write, execute (or cd to it if it is a directory)
- First symbol is either d or – (directory or file), next three user permissions, next three group permissions, last three other's permissions.

```
drwxr-xr-x 2 minas faculty   4096 Mar  6 08:47 zip
-rw-r--r-- 1 minas faculty    379 Jan  9 22:51 wordcnt.c
```

File permissions

- You can change file permissions with the chmod command
  - Eg: chmod a+r notes.html
    - Gives permission to all to read
  - chmod g-r notes.html
    - Withholds permission from group to read.

File Permissions

- There are other permissions:
  - Set-uid: when file is executable, the user id it the owner of the file. Examples: passwd, shutdown
  - Set-gid: when file is executable the group ID is the group ID of the file
  - Sticky bit: Prevents one user from deleting another user’s /tmp files.
- Used far less often

Process management

- We can examine the processes in Linux (but not many Unix variants) through /proc.
- Also using the commands jobs, ps, kill, killall, bg, fg, sleep, etc.
- We can kill a process with ^C, suspend a process with ^Z (notice the difference from MS-Windows)

File System Mounts

- Command df gives info about file system mounts:
  - Name of physical disk (could be through the internet, or a special filesystem)
  - Total, used and available capacity in blocks.
  - Percentage used
  - Path of the mount point
- Command du gives sizes of subdirectories (help user find huge and wasteful files)
- Command file indicates the nature of the file
  - Ie text, postscript, pdf, executable, etc

Variables

- Different shells differ on how they treat variables
- Many are already defined and help you run your programs
  - echo $PATH
  - echo ${PATH}
  - echo ${SHELL}
  - printenv|less

Variables

- Assigning (setting) variables
  - varname=value
  - demovar=2031
  - echo demovar
  - echo $demovar
  - echo ${demovar}
  - echo ${demovar}N
  - echo $demovarM

Shell vs Environment Variables

- Shell variables are local to the shell and do not affect programs run by the shell
- Environment variables are exported to the program run by the shell
  - Programs can access them with getenv(3)
    - Ie man -s 3 getenv
- You can export variable with
  - export varname
  - declare -x varname
- You can unexport them with
  - export -n varname

```
#!/bin/bash

filepath=lab5-marked/${REMOTE_USER}.tar
if [ -f $filepath ] ; then
    echo 'Content-description: File Transfer'
    echo 'Content-type: application/octet-stream'
    echo Content-length: `cat ${filepath}|wc -c`
    echo Content-Disposition: attachment\; filename=\"${REMOTE_USER}.tar\
    echo
    cat ${filepath}
    exit 0
else
    echo 'Content-type: text/html'
    echo
    echo '<!DOCTYPE html>'
    echo '<html>'
    echo '<head>'
    echo '<title>Mark not Available</title>'
    echo '</head>'
    echo '<body>'
    echo '<p>No mark available</p>'
    echo '</body>'
    echo '</html>'
fi
```