Absolute path: starts at the root “/”. Relative path starts at the current directory

Permissions: Read, Write, Execute

Setuid: execute with permissions of the user

Setgid: execute with permissions of the group

Stickybit: Adds a rule to directory (only owner of file can delete a file)

* Deleting a file is dependent on directory permissions (not file permissions)

Chmod: Changes pemissions ex: chmod 711, etc.

Hard Links vs Soft Links: Hard link is another reference to the inode. A soft link is like a shortcut, it is a pointer to a file (points to a name of a file, not the inode)

Bourne Shell: Standard shell (Bash is a superset of bourne shell)

Running commands in background: Use ‘&’ (after command) (gives you your output immediately, but gives you back control of the shell immediately, aka while the program is running)

I/O redirection: command > file (wipes out file if exists, creates if doesn’t and puts output in it)

Command >> file (appends to end of file if exists, creates if doesn’t and puts output in it

Ex: date >/tmp/foo 2>&1 (send standard error to the same place you are sending standard output, which you just set to /tmp/foo

AFTER EXAM 1:

#print sums for each of n columns plus an overall total

awk ‘

BEGIN { n = ‘$1’ }

{

For (i = 1; i <= n; i++) {

Printf “%4.1f “, sum[i]

Total += sum[i]

}

Printf “; total = %4.1f\n”, total

}’ $2

# $1 is first argument to shell

# $Home = ~ = home directory (this is an environment variable)

# <$Home/calendar.txt <- this means get input from calendar.txt (located in the home directory)

#`date` causes shell to run command date and replace date in script with output