**CS 3421 Week 2 Problems: File Name Patterns, Aliases, and chmod**

**1 File Name Patterns**

Assume the following files existing in the working directory:

1input.txt 2output.txt A1 Assign1.h Assign2.h output.txt

1output.txt 3output.txt A2 Assign1.o Assign2.o

2input.txt 6output.txt Assign1.c Assign2.c input.txt

1a: Use ls and file name patterns to list all files belonging to assignment 1 (i.e., all of those with the number “1” in the filename).

ls \*1\*

1b: Use ls and file name patterns to list all .c files.

ls \*.c

1c: List all files beginning with the letter A

ls A\*

1d: List all files beginning with any number between 0 and 4.

ls [0-4]\*

1e: List all files with either the word “input” or “output” in the filename.

ls \*{input,output}\*

1f: List all files beginning with “A” which end with an extension.

ls A\*.\*

1g: List all files beginning with “A” followed by only one character.

ls A?

1h: List all files not beginning with a number.

ls [^0-9]\*

**2 Aliases**

2a: Create an alias, numberfiles, which prints the number of files in the current directory

alias numberFiles=’ls –A | wc –l’

-A will show visible + hidden(dot) files

-a will show –A plus the prev and parent directory

2b: Using an alias, replace ls so that it now always shows hidden files.

alias ls=’ls –A’

2c: Create an alias, fox1, which uses ssh to attempt to log you into fox01.cs.utsa.edu and prompt you for your password.

alias fox1=’ssh –X abc123@fox01.cs.utsa.edu’

2d: Assuming the variable $var is set to 100, create an alias, whatIsVar, which echos the *current value* of $var. Note that if $var changes to another value, whatIsVar should reflect that automatically.

alias whatIsVar=’echo $var’

single quotes cause alias to interpret value each time it’s executed. Double quotes will only interpret the value when the alias was created.

2e: Solve the same problem above, except make it so that whatWasVar always prints the original value of $var (i.e., if $var changes, whatIsVar continues to print 100).

alias whatIsVar=”echo $var”

**3 chmod**

3a: Using symbolic modes, update file1 to remove write access from users who are not the owner or belong to the file’s group.

chmod o-w file1

3b: Using symbolic modes, update file1 so that group and other have only read access

chmod go=r file1

3c: Using symbolic modes, update all files in the directory so the owner has execute access to them

chmod u+x \*

3d: Using symbolic modes, update all files in the directory ending with .jar to *add* execute access for the owner and set group and other to have *only* read and execute access

chmod u+x,go=rx \*.jar

3e: Using octal notation, update the file1 to have *only* read and write access for the owner

chmod 600 file1

3f: Using octal notation, update file1 to have read, write, and execute access for the owner, read and write access for the group, and read access for everyone else.

chmod 764