

CST8177 – Linux II

Midterm Solution

more shell

Todd Kelley

kelleyt@algonquincollege.com

Today's Topics

- ▶ midterm solution
- ▶ debugging shell scripts
- ▶ exit
- ▶ case
- ▶ is stdin a terminal?
- ▶ the command that does nothing
- ▶ integer arithmetic

Debugging shell scripts

- ▶ `-v` option for `bash/sh`
 - `sh -v myscript`
 - shell will print each line as its read
 - loop statements are printed once
- ▶ `-x` option for `bash/sh`
 - `sh -x myscript`
 - shell will display `$PS4` prompt and the expanded command before executing it
 - each loop iteration is shown individually

exit command

- ▶ `exit` causes the shell to exit with the exit status of the last command that was run
- ▶ `exit N` causes the shell to exit with exit status `N`

case statement

```
case test-string in
  pattern-1)
    command1
    command2
    ;;
  pattern-2)
    command3
    command4
    ;;
  *)
    command5
    ;;
esac
```

case statement continued

- ▶ the patterns are globbing patterns matched to the test-string
- ▶ So we tend to use the * pattern as a catchall, if all other matches fail, but that's not required
- ▶ case statement exit status is the exit status of the last command in the matching block, or 0 if no blocks match

case statement continued

- ▶ We can use the vertical bar to specify alternative patterns:

```
case "$character" in
    a|A)
        echo "The character is A"
        ;;
    [bB])
        echo "the character is B"
        ;;
    *)
        echo "The character is not A or B"
        ;;
esac
```

Is stdin a terminal?

- ▶ A script can test whether or not standard input is a terminal

```
[ -t 0 ]
```

- ▶ What about standard output, and standard error?

The command that does nothing

- ▶ Occasionally you'll see a command called :
- ▶ : arguments
- ▶ That command expands its arguments and does nothing with them, resulting in a 0 exit status

Doing integer arithmetic

- ▶ examples of using `expr` command:

```
a=`expr 3 + 4`
```

```
a=`expr 3 - 4`
```

```
a=`expr 3 * 4`
```

```
a=`expr 13 / 5` # integer division: 2
```

```
a=`expr 13 % 5` # remainder: 3
```

- ▶ increment the integer in variable `a`

```
a=`expr $a + 1`
```