CST8177 – Linux II

More Scripting

Todd Kelley
kelleyt@algonquincollege.com

Today's Topics

- trailing topics from last week
- Scripting reference material
- more about test program
- numbers versus strings
- || and &&

Scripting Reference Material

- Sobel Chapter 27
- http://teaching.idallen.com/cst8129/05f/notes/exit_status.txt
- http://teaching.idallen.com/cst8129/05f/notes/quick_tests.txt
- http://teaching.idallen.com/cst8129/05f/notes/quotes.txt
- http://teaching.idallen.com/cst8129/05f/notes/script_checklist.txt
- http://teaching.idallen.com/cst8129/05f/notes/script_style.txt
- http://teaching.idallen.com/cst8129/05f/notes/shell_read.txt
- http://teaching.idallen.com/cst8129/05f/notes/shell_script_execution.txt
- http://teaching.idallen.com/cst8129/05f/notes/shell_variables.txt

Why scripting?

Why would anyone want to know how to write a script?

Why is is particularly important for a sysadmin?

Here are 7 reasons to consider:

1. avoid complex typing, preventing possible errors
2. automate repetitive tasks
3. use when an alias gets too complex or not possible
4. make new, specialized commands
5. automate long and/or complex tasks
6. handle rare but complex activities
7. create a "wrapper" for a program

These are all valid reasons, especially for a sysadmin managing a Linux/Unix server on behalf of an enterprise. The reasonable use of scripting will make you more productive, more accurate, and more efficient, increasing your value to your employer.

Production Examples

- gunzip
  - vi `which gunzip`
- vimtutor
  - vi `which vimtutor`

test program examples

- man test (three categories of simple test)
  - tests about files
    - is it a file or a directory
    - is it executable?
    - does it have its SUID bit set?
  - tests about strings
    - is it a null string?
    - is one string equal to another?
    - is one string alphabetically before or after?
  - tests about numbers?
    - equal to, less than, greater than?

combining tests together

- building complex tests from simple tests
- test1 -a test2
  - both test1 and test2 must be true
- test1 -o test2
  - at least one must be true
- ! test1
  - test1 must be false
- (test1)
  - true if test1 is true

numbers versus strings

- numbers and strings are not the same
- "00" is not the same as "0" (as strings)
  - one has two characters, the other has one character
  - how can they be the same?
  - if your pin number is 0037 and someone tries, 37, should it work?
  - as strings, "0037" is not the same as "37" so, no, it shouldn't work
- = and != and < and > are for STRINGS

Numbers versus Strings

- numbers and strings can't be mixed
- 00 is the same as 0 (as numbers)
- 005 is the same as 05 and 5 (as numbers)
- it's an error to ask if the number 0 is the same as the number xyz (error: there is no such number xyz)
- -eq -ne -gt -lt -ge -le are for NUMBERS
- `0 -eq xyz` #gives ERROR, xyz not a NUMBER

|| and &&

- We have already seen `&&` in action:
  - `[ -f $HOME/.bashrc ] && . $HOME/.bashrc`
  - `[ -z "$PS1" ] && return`
- And we've seen these are equivalent to

```
if [ -f $HOME/.bashrc ]; then
     . $HOME/.bashrc
fi
```

and

```
if [ -z "$PS1" ]; then
     return
fi
```

&& means "and"

- Suppose you might qualify for a scholarship:
- Those who qualify are:
  - eight feet tall, and ??
  - born on the moon, and ??
  - algonquin student and ??
- or in other words
  - eight feet tall `&&` ??
  - born on the moon `&&` ??
  - algonquin student `&&` ??
- In which case do we need to find out what ?? is?

&& continued

- With "and", we need to keep going as long as we keep encountering "true"
- As soon as we encounter "false", we can stop
  - born on the moon && ?? # we don't care about ??
  - algonquin student && ?? # we need to know ??
- In the first case, we would not do the ?? command, whatever it is
- In the second case, we would do the ?? command, whatever it is
- Often we don't need the exit status of the command ??, we just wanted the command to run (or not)

|| means "or", opposite of &&

- As soon as we encounter "true", we can stop
- You qualify for a $1000 rebate under the following conditions:
  - born on the moon, or ??
  - algonquin student, or ??
- In the first case, we need to know what the exit status of the ?? is, we need to run the ?? command
- In the second case, we can stop before running the ?? command

&& and || in general

- aa && bb
  - means if aa; then bb; fi

- aa || bb
  - means if not aa; then bb; fi

Checking Lottery Numbers

- You have six numbers, and six numbers are drawn randomly
- When you're checking the numbers, is it && or || that governs when you stop checking?

first number matches &&
second number matches &&
third number matches &&
..etc...

Is your name on a list?

- After you try out for the basketball team, the list of people who made the team is posted.
- When you're checking for your name on the list, which of && or || governs when you stop checking?

first name on the list is yours ||
second name on the list is yours ||
third name on the list is yours ||
...etc...

&& versus -a, || versus -o

- && and -a both mean "and"
  - && is used between commands
  - -a is used between expressions in the test command

- || and -o both mean "or"
  - || is used between commands
  - -o is used between expressions in the test command

&& and || versus -a and -o

- && and || are used with commands that tend to get things done
  - to graduate, you
    - complete first year && complete second year
  - complete first year is a "command" that gets things done: you learn the first-year material
- -a and -o are used in test, and don't do things, just affect the exit status of test
  - you are a rich canadian if
    - you are canadian -a you are rich
  - checking whether or not you're canadian doesn't get things done - but it does establish a truth value

Rich Canadian

- If we used &&, we'd be getting things done
- become a Canadian && earn $1000000
- When it comes to trying to be a rich Canadian, if we fail to become Canadian, we don't need to bother earning $1000000