Scripting and Computer Environments

IIIT-H

Sep 1, 2012

...So Far & Today...

- Basic Linux Commands
- Filters (both simple and power)
- Regex
- Basic Shell Scripting
  - Shell Metacharacters
  - Expressions
  - Flow Control
    - Branching
    - Looping
- Today:
  - Q &A / problem solving
  - Functions
  - Debugging Scripts

...So Far & Today...

- Basic Linux Commands
- Filters (both simple and power)
- Regex
- Basic Shell Scripting
  - Shell Metacharacters
  - Expressions
  - Flow Control
    - Branching
    - Looping
- Today:
  - Q &A / problem solving
  - Functions
  - Debugging Scripts

Expectations: Basic Linux Commands

- Ofcourse, their purpose.
- Common flags
- Common File operations
- Redirection and piping

Expectations: Filters

- What are they? Which ones?
- Usage with/without regex
- Regex metacharacters and their uses
- Write a regex / interpret a regex

- Components of a typical script
- Usage of basic constructs (`test, if, case, for, while, ...`)
- Write small scripts

- Display the largest five files in the current directory. Save the new list to a file named `LargestFive.txt`

  Use `ls -l`

Examples:

- Matching regex for:

  `heard, hard, herd`
  `cool, coolant, cooler, coolest, coolness`

- Any difference between

  `grep ^[^A-K] MyFile` and
  `grep -v ^[A-K] MyFile ?`

- Display all words of the file `/usr/share/dict/words` which are of length exactly 15 characters. How about those of length 20 characters and beyond?

- Regex to match a date in the format dd/mm/yy.

Examples:

- Matching regex for:

```
heard, hard, herd
cool, coolant, cooler, coolest, coolness
```

- Any difference between

```
grep ^[^A-K] MyFile                  and
grep -v ^[A-K] MyFile ?
```

- Display all words of the file `/usr/share/dict/words` which are of length exactly 15 characters. How about those of length 20 characters and beyond?
- Regex to match a date in the format dd/mm/yy.

Examples: Regex

- Matching regex for:

  ```
  heard, hard, herd
  cool, coolant, cooler, coolest, coolness
  ```

- Any difference between

  ```
  grep ^[^A-K] MyFile               and
  grep -v ^[A-K] MyFile ?
  ```

- Display all words of the file `/usr/share/dict/words` which are of length exactly 15 characters. How about those of length 20 characters and beyond?

- Regex to match a date in the format dd/mm/yy.

- Matching regex for:

```
heard, hard, herd
cool, coolant, cooler, coolest, coolness
```

- Any difference between

```
grep ^[^A-K] MyFile                and
grep -v ^[A-K] MyFile ?
```

- Display all words of the file `/usr/share/dict/words` which are of length exactly 15 characters. How about those of length 20 characters and beyond?

- Regex to match a date in the format dd/mm/yy.

Decoding sed commands:

```
sed -n "1,/^$/p"
```

```
sed "/^ya*y/,/[0-9]$/d"
```

```
sed -E "s/([0-9])-([0-9])/ \1 \2 / g" file.txt
```

Examples: Filters: sed (2)

Write a sed command:

- `cat test.txt` displays full names in the form of : `firstname:middlename:lastname`
  1. Output all names as `lastname:middlename:firstname`
  2. Now, as `firstname_initial.middlename_initial.lastname` E.g. `Alice:Bob:Charlie` → `A.B.Charlie`
- Common pattern filtering (e.g. SSN filtering, license plate, mobile numbers ...)

Examples: Filters: sed (2)

Write a sed command:

- `cat test.txt` displays full names in the form of : `firstname:middlename:lastname`
    1. Output all names as `lastname:middlename:firstname`
    2. Now, as `firstname_initial.middlename_initial.lastname`
       E.g. `Alice:Bob:Charlie` → `A.B.Charlie`
- Common pattern filtering (e.g. SSN filtering, license plate, mobile numbers ...)

Examples: Filters: sed (2)

Write a sed command:

- `cat test.txt` displays full names in the form of : `firstname:middlename:lastname`
    1. Output all names as `lastname:middlename:firstname`
    2. Now, as `firstname_initial.middlename_initial.lastname`
       E.g. `Alice:Bob:Charlie` → `A.B.Charlie`
- Common pattern filtering (e.g. SSN filtering, license plate, mobile numbers ...)

Examples:

1

2

3

4

(Refer to the example scripts sent)

Examples: Shell Scripting

1.
2.
3.
4.

(Refer to the example scripts sent)

Examples:

1.
2.
3.
4.

(Refer to the example scripts sent)

Examples: Shell Scripting

1.
2.
3.
4.

(Refer to the example scripts sent)

Shell Functions

- A function is a named section of a code that can be executed.
- Two parts: function body + function arguments.
- May return a value.

Format

```
function_name()
{
statements
return <value>                                   #optional
}
```

Data Validation

Thou shalt validate thy data/input!

Debugging Shell Scripts

- The process of ascertaining why the script is not working.
- Finding and reducing the number of bugs in a script.
- Origin: Grace Hopper's moth bug.
- Some debuggers: gdb, pdb (Python), Idb, dbx, visual debuggers (e.g. MS Visual Studio), etc

```
bash -x <scriptname>
```