Scientific and Techincal Computing: SMR3821

Introduction to Linux Text Editing and Shell Scritping.

Elliot S. Menkah, Ph.D

National Institute for Mathematical Sciences, Ghana.
Kwame Nkrumah University of Science and Technology, Ghana.
ICTP, Trieste.

November 7, 2022

Linux - Introduction to Text editing and Shell Scripting

Text Editing and Shell Scripting

Introduction to Shell Scripting

1. Editing with Linux text editors
    1. Nano
    2. Vi or Vim
    3. Emacs
2. Bash Shell Scripting

Linux - Introduction to Text editing and Shell Scripting

Nano Syntax & Structure

To starting text editing with Nano:

```
~ $ nano <file-name>
```

After adding text content to the file

Editing operations of Nano

ctrl + O - Write to file(save changes made)
ctrl + X - Close the opened file
ctrl + G - Get help with Nano
ctrl + W - Search or find a string in text
...
...

Vim Syntax & Structure

To starting text editing with Vi or Vim:

```
1 ~ $ vim <file-name>
```

Def: Escape mode

Modes of Vi/Vim

- Escape mode - esc key
- INSERT mode - i key
- VISUAL Block mode - ctrl + v

Linux - Introduction to Text editing and Shell Scripting

Vim Syntax & Structure

To starting text editing with Vi or Vim:

```
1 ~ $ vim <file-name>
```

After adding text content to the file, get into ESC mode

Editing operations of Vim

w - Write to file(save changes made)
q - quit vim of close the opened file

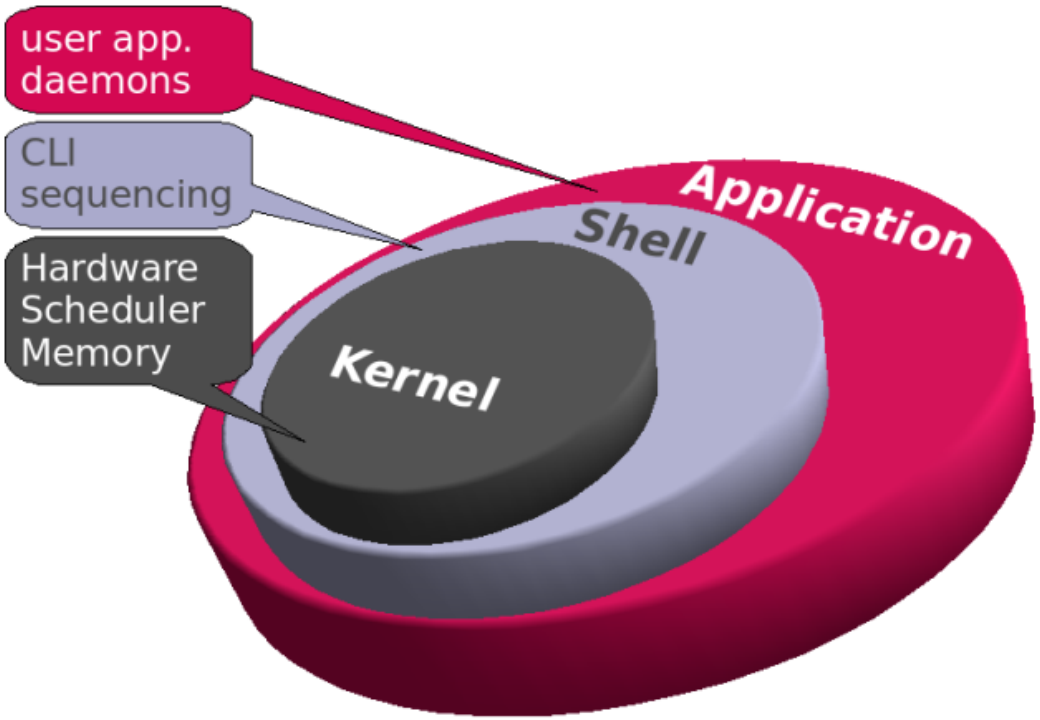
Shell Scripting

Introduction to Shell Scripting

1. Shell Scripting

Linux - Introduction to Text editing and Shell Scripting

Computer Structure



Ref to image: Kernel & Shell.

- Shells :
  - Borne Shell
  - Borne-Again Shell(Bash)
  - korn shell
  - C shell

  ...

  ...

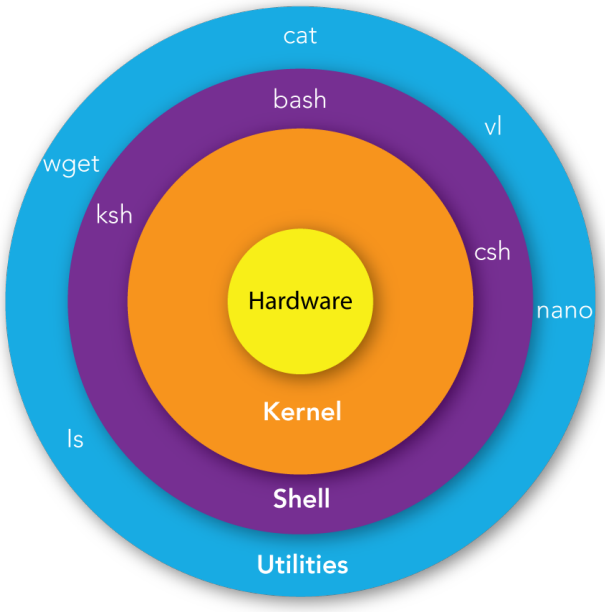
Linux - Introduction to Text editing and Shell Scripting

Shell scripts & The Computer Structure.

Why shell scripts look like.

```
#! /bin/bash

echo `date` > myfile.txt
echo "Hello There" > myfile.txt

echo "My first Shell script" >> myfile.txt

mkdir -p scripthandson

mv myfile.txt scripthandson
```

Computer Structure.



Ref to image: Kernel & Shell.

Linux - Introduction to Text editing and Shell Scripting

Why is shell scripting even necessary ?

- Importance:
  - Writing a series of commands
  - Combine lengthy and repetitive commands
  - Execute Routine task

  ...

  ...

Linux - Introduction to Text editing and Shell Scripting

How to create a shell script

- Steps:
  1. Create a file(with your preferred text editor) and name it with a .sh extension.
  2. Start the content of the script with #!(shebang) /path/to/shell/.
  3. Add some code/text/content to the file/script and save.
  4. Modify file permissions of script to make it executable .

```
#! /bin/bash

echo `date` > myfile.txt
echo "Hello There" > myfile.txt

echo "My first Shell script" >> myfile.txt

mkdir -p scripthandson

mv myfile.txt scripthandson
```

Linux Command Line - Shell Scripting & Access Control

chown:: Change ownership of files

chmod: Change permission on files

setuid: Share ownership on files

sticky bit: Share write access on a directory

Making file executable

To change the permission to make file executable by user:

```
~ $ chmod u+a <script-name.sh>
```

Running executable script

To run or execute script:

```
~ $ ./<script-name.sh>
```

or

```
~ $ bash <script-name.sh>
```

Linux Command Line - Shell Scripting & Access Control

- **Comments**
- **Variables**
- **Statements**
- **Conditionals**
- **Controls sequence/ Loops**
- **Functions**

Comments in Scripting

Comments in shell scripting are denoted with a preceding # symbol.

Comments

```
#! /bin/bash

# Illustration of comments in shell scripting
# Author: Elliot Menkah
# Email: elliotsmenkah@gmail.com

echo `date` > myfile.txt
echo "Hello There" > myfile.txt

echo "My first Shell script" >> myfile.txt

mkdir -p scripthandson

mv myfile.txt scripthandson
```

Shell Variables

Shell Variables store data.

```
#! /bin/bash

# Illustration of comments in shell scripting
# Author: Elliot Menkah
# Email: elliotsmenkah@gmail.com

fname='Elliot'

echo `date` > myfile.txt
echo "Hello There" > myfile.txt

echo "My firstname is $fname" >> myfile.txt

echo "This is my first Shell script" >> myfile.txt

mkdir -p scripthandson

mv myfile.txt scripthandson
```

;

Conditionals

Conditionals are tools for decision making.

```
#! /bin/bash

# Illustration of comments in shell scripting
# Author: Elliot Menkah
# Email: elliotsmenkah@gmail.com

echo `date` > myfile.txt

echo "This is my first Shell script" >> myfile.txt

num1=5
num2=2

if [ $num1 -gt $num2 ]; then
    echo "$num1 is greather than $num2"
else
    echo "$num2 is greather than $num1"
fi
```

Control Sequence/ Loops

Control Sequence or loops are used to iteratively parse instructions to be executed.

```
#! /bin/bash

# Illustration of control sequence with for loops in shell
# scripting
# Author: Elliot Menkah
# Email: elliotsmenkah@gmail.com

echo `date` > myfile.txt

for i in 1 2 3;
do
    echo $i;
done;


for i in $(seq 1 10);
do
    echo $i;
done;
```

Functions

A functions is a way or technique for grouping reusable bits of code under one name for later use.

```
#! /bin/bash

# Illustration of functions in shell scripts
# Author: Elliot Menkah
# Email: elliotsmenkah@gmail.com

echo `date` > myfile.txt

my_print_func(){
    echo "Hi there, this is my simple print function"

}

my_sum_func(){

    res=$(($num1 + $num2))
    echo "Sum of $num1 and $num2 = $res"
    return $res

}

print_func
my_sum_func
```

End of talk

Thank you ...