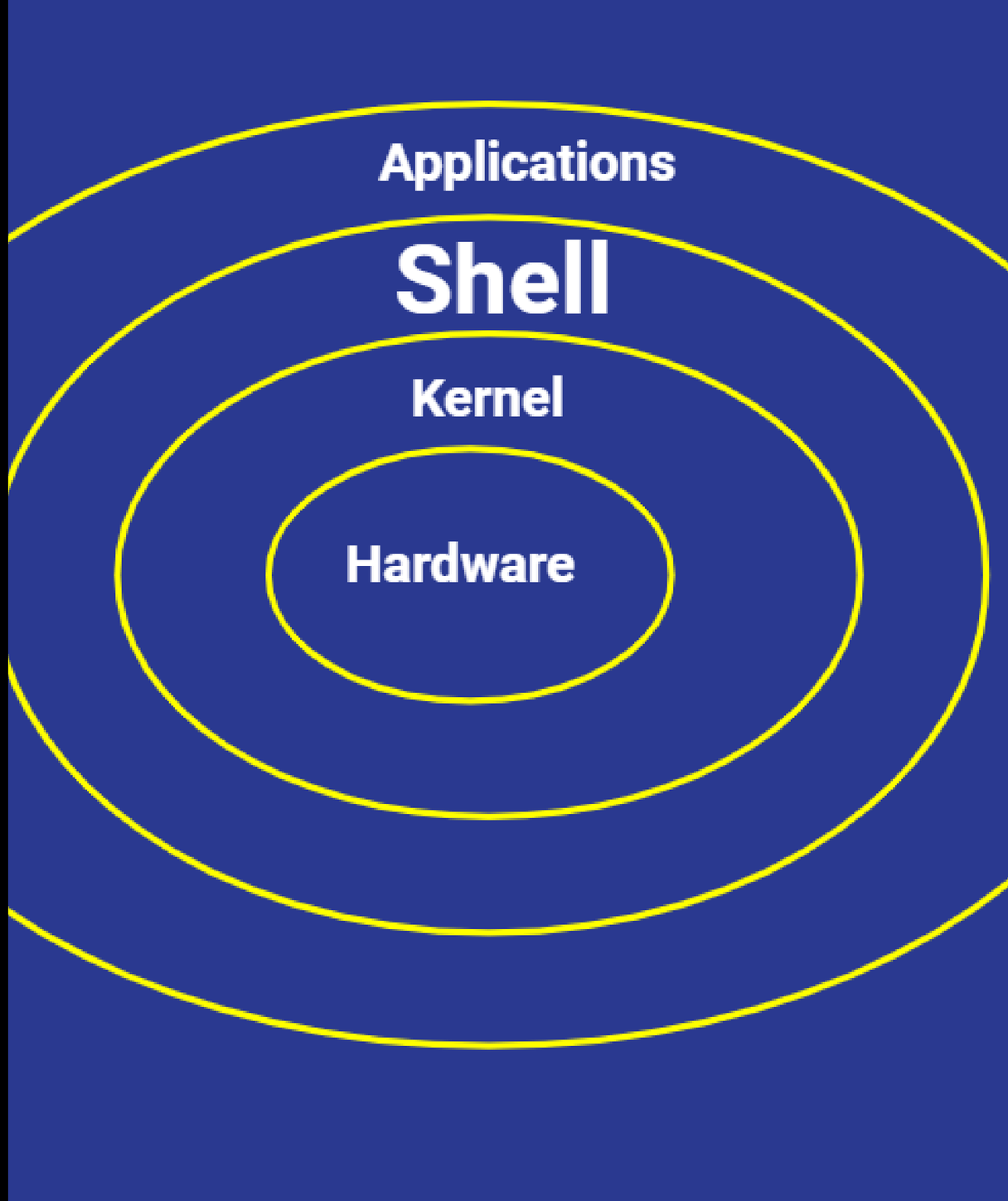
MPRASHANT

SHELL SCRIPTING

MPRASHANT

WHAT IS LINUX SHELL?

A shell provide an environment to a user to execute commands and interact with kernel.



There are different types of shell

- bash
- sh
- ksh
- tsh
- fish
- zsh

MPRASHANT

WHAT IS MY SHELL TYPE?

You can check using

```
echo $0
```

MPRASHANT

WHAT IS SHELL SCRIPTING?

- Shell script consist of set of commands to perform a task.
- All the commands execute sequentially.
- Some task like file manipulation, program execution, user interaction, automation of task etc can be done

MPRASHANT

FIRST BASIC SCRIPT...

```
#!/bin/bash
echo "Hello World!"
```

MPRASHANT

WHAT IS SHEBANG?

```
#!/bin/bash
```

MPRASHANT

SENDING OUTPUT TO TERMINAL..

```
echo "Hello World!"
```

MPRASHANT

HOW TO RUN A SCRIPT..

- Make sure script has execute permission `rwx`

- Run using
  - `./script.sh`
  - `/path/script.sh`
  - `bash script.sh`

- Ctrl+C to terminate
- Ctrl+z to stop

MPRASHANT

COMMENTS

Using #

```
#This is a comment
```

Multi-line comment

```
<<comment
...
your comment here
...
comment
```

MPRASHANT

WHAT ARE VARIABLES?

```
VAR_NAME=value

VAR_NAME=$(hostname)


echo $VAR_NAME
```

MPRASHANT

CONSTANT VARIABLE?

Once you defined a variable and don’t wanna change it until end of the script.

```
readonly var_name=”Hi”
```

Arrays

MPRASHANT

ARRAYS

How to define an array?

```
myArray=( 1 2 Hello "Hey man")
```

How to get values from an array?

```
echo "${myArray[0]}"
echo "${myArray[1]}"
```

MPRASHANT

ARRAYS

How to get length of array?

```
echo "${#myArray[*]}"
```

How to get specific values?

```
echo "${myArray[*]:1}"
echo "${myArray[*]:1:2}"
```

MPRASHANT

ARRAYS

How to update an array?

```
myArray+=( 5 6 8 )
```

MPRASHANT

ARRAYS
KEY-VALUE

```
declare -A myArray
myArray=( [name]=Paul [age]=20 )

echo "${myArray[name]}"
```

String Operations

MPRASHANT

STRING OPERATIONS

```
myVar="Hello World!"

length=${#myVar}

upper=${x^^}

lower=${y,,}

replace=${myVar/World/Buddy}

slice=${myVar:6:11}
```

User Interaction

MPRASHANT

TAKING INPUT FROM USER...

```
read <var_name>

read -p "Your name" NAME
```

Arithmetic Operations

MPRASHANT

HOW TO USE EXPRESSIONS

Using let command

```
let a++
let a=5*10

((a++))
((a=5*10))
```

Conditional Statement

MPRASHANT

IF-ELSE

```
if [ $marks -gt 40 ]
then
        echo "You are PASS"
else
        echo "You are FAIL"
fi
```

MPRASHANT

OPERATORS

Equal	-eq / ==
Greaterthanorequalto	-ge
Lessthanorequalto	-le
Not Equal	-ne / !=
Greater Than	-gt
Less Than	-lt

MPRASHANT

ELIF

```
if [ $marks -ge 80 ]
then
        echo "First Division""
elif [ $marks -ge 60 ]
then
        echo "Second Division"
else
        echo "Fail""
fi
```

MPRASHANT

CASE

```
echo "Hey choose an option"
echo "a = To see the current date"
echo "b = list all the files in current dir"

read choice

case $choice in
        a) date;;
        b) ls;;
        *) echo "Non a valid input"
esac
```

Logical Operators
&&, ||, !

MPRASHANT

LOGICAL OPERATORS

```
condition1 && condition2
```

If both conditions are true then true else false

```
condition1 || condition2
```

If any of the condition is true then true

MPRASHANT

LOGICAL OPERATORS

```
condition1 && condition2 || condition3
```

Execute condition2 only when 1 is true else execute condtion3

Loops

MPRASHANT

FOR LOOP

```
for i in 1 2 3 4 5
do
        echo "Number is $i"
done
```

Other ways to write For loop

```
for j in Raju Sham Baburao
for p in {1..20}
```

MPRASHANT

ITERATE VALUES FROM FILE..

```
items="/home/paul/file.txt"

for item in $(cat $items)
do
        echo $item
done
```

MPRASHANT

WHILE LOOP

```
count=0
num=10

while [ $count -le $num ]
do
        echo "Numbers are $count"
        let count++

done
```

MPRASHANT

UNTIL LOOP..

```
a=10

until [ $a -eq 1 ]
do
    echo $a
    a=`expr $a - 1`
done
```

MPRASHANT

INFINITE LOOP..

```
while true
do
        echo "Hi'
        sleep 2s
done
```

MPRASHANT

To read content from a file

```
while read myVar
do
        echo $myVar

done < file_name
```

MPRASHANT

WHILE LOOP

To read content from a csv file

```
while IFS="," read f1 f2 f3
do
        echo $f1
        echo $f2
        echo $f3
done < file_name.csv
```

MPRASHANT

WHILE LOOP

To read content from a csv file

```
cat myfile.csv | awk '!NR=1 {print}'
| while IFS="," read f1 f2 f3
```

Functions

MPRASHANT

WHAT ARE FUNCTIONS?

- Block of code which perform some task and run when it is called.
- Can be reuse many times in our program which lessen our lines of code.
- We can pass arguments to the method

MPRASHANT

HOW TO MAKE FUNCTIONS?

```
function myfun {
    echo "Hi"
}

myfun() {
    echo "Hello"
}
--------------------------------

To call the function
myfun
```

MPRASHANT

HOW TO USE ARGUMENTS IN FUNCTIONS?

```
addition() {
        local num1=$1
        local num2=$2
        let sum=$num1+$num2
 echo “Sum of $num1 and $num2 is $sum”
}

--------------------------------

myfun 12 13
```

Arguments Passing

MPRASHANT

ARGUMENTS IN SCRIPT...

```
#myscript.sh arg1 arg2..

How to access these arguments inside
our script?

To get no. of arguments    : $#
To display all arguments    : $@
To use or display a argument: $1 $2 ..
```

MPRASHANT

ARGUMENTS IN SCRIPT...

```
for arg in $@
do
        echo "Argument is $arg"
done
```

MPRASHANT

SHIFTING ARGUMENTS

SHIFT

When we pass multiple arguments, we can shift.

```
A B C
shift
B C
```

Other Useful Concepts

MPRASHANT

USEFUL CONCEPTS...

break - to stop the loop

continue - to stop current iteration of loop and start next iteration

MPRASHANT

USEFUL CONCEPTS...

`sleep` - to create delay between two executions ex: sleep 1s/1m

`exit` - to stop script at a point

exit status `$?` - gives you status of previous command if that was successful

MPRASHANT

USEFUL CONCEPTS...

basename - strip directory info and only give filename

dirname - strip the filename and gives directory path

realpath - gives you full path for a file

MPRASHANT

CHECK IF FILE/DIR EXIST

```
if [ -d folder_name] If folder exists

 [ ! -d folder_name] If folder not exists
```

```
if [ -f file_name] If file exists

if [ ! -f file_name] If file not exists
```

MPRASHANT

BASH VARIABLES...

RANDOM - A random integer between 0 and 32767 is generated

UID - User ID of the user logged in

Redirection in scripts

> >>

MPRASHANT

WHAT IS /DEV/NULL

In case if you don’t wanna print the output of a command on terminal or write in a file,

We can redirect the output to /dev/null

Example:

```
#cd /root &> /dev/null
```

MPRASHANT

PRINT NAME OF THE SCRIPT

```
echo "The name of the script is: ${0}"
```

MPRASHANT

LOG MESSAGES..

If you want to maintain the logging for your script, you can use logger in your script.

You can find the logs under /var/logs/messages

Example: #logger “Hey Buddy”

MPRASHANT

DEBUGGING SCRIPTS

If We can enable the debugging of the script using below in the script

```
set -x
```

MPRASHANT

DEBUGGING SCRIPTS

If We want to exit our script when a command fail

```
set -e
```

Running Script in background

```
nohup ./script.sh &
```

Automate our Script

At or Crontab

For scheduling only one time, use AT

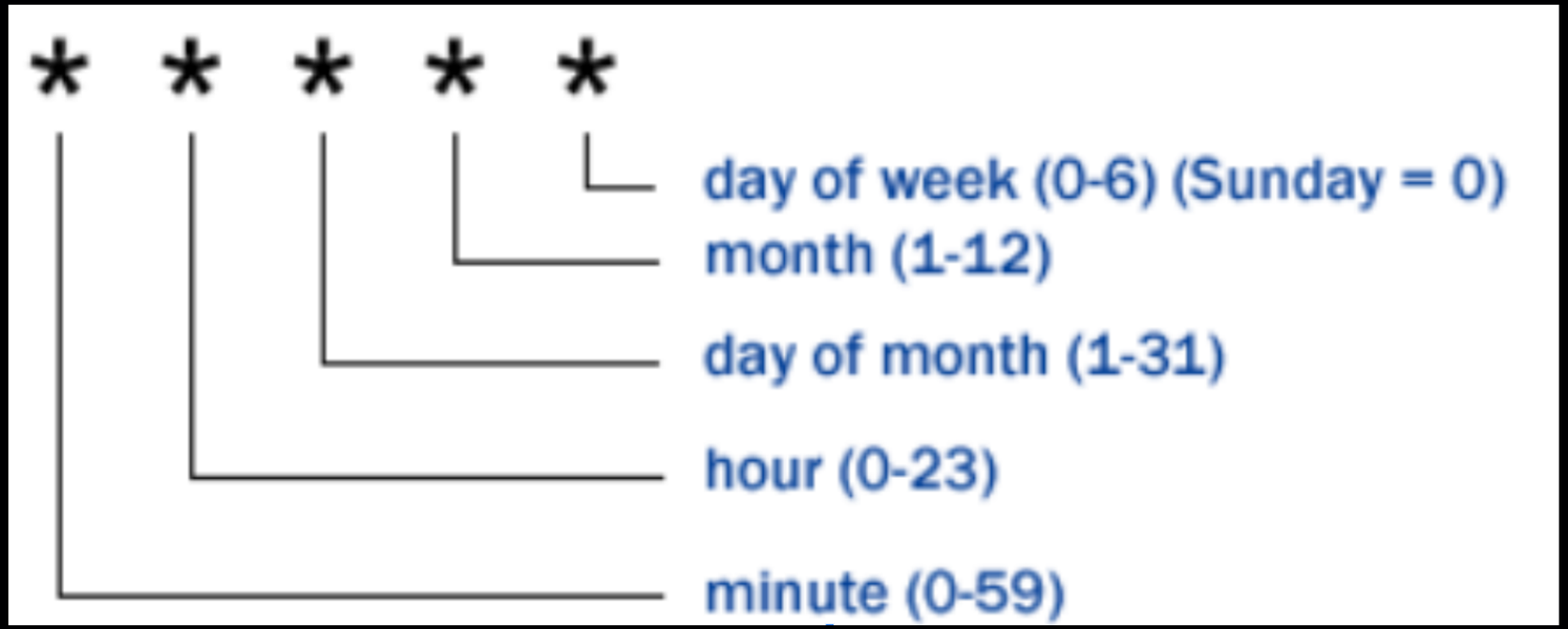
at 12:09 PM
  <your_command>
Ctrl + D

atq to check scheduled job
atrm <id> to remove the schedule

To check the existing jobs - crontab -l

To add new job - crontab -e

```
* * * * cd /home/paul/scripts && ./create_file.sh
```


* * * * *
day of week (0-6) (Sunday = 0)
month (1-12)
day of month (1-31)
hour (0-23)
minute (0-59)

Projects

MPRASHANT

1 Monitoring free RAM space

MPRASHANT

2

Monitoring free DISK space and sent an Alert Email

MPRASHANT

3

Access Remote server and perform some actions

MPRASHANT