

Command

Kernel version check: uname -r

Architecture : uname -m (32 / 64)

Print working directory: pwd

List of current dir: ls

Hidden files

Clear: screen clear

make directory: mkdir os

"multiple": mkdir A B C D

Change directory: cd A [inside A directory]

Back to previous folder: cd .. [Parent folder back]

Return to home dir: cd

Home directory folders: cd / home / student

Actual path: Desktop / A / A1 [Actual path]

cd ~ / Desktop / A / A1 [relative path]

empty file make filename

make file touch f1 [by default text file]

Write inside file: cat &gt; f1 → Ctrl + D [file terminate]

Read file: cat f1

Directory

`cp renamedfile -/desktop/LabP1/dir2`

automatic file create কোরি ফাইল মানে জাইম: `cat > f2`

append 2 file: `cat f1>>f2` [f2 update হবে]

2'th file কে concat কোরি তৈরি file দ্বাৰা: `cat f1 f2>f3`

copy file: `cp f3 f4` [f3 source, f4 destination]

f3 desktop এ, f4 অন্য folder ও মানে path ফাইল হবে,

rename file: `mv f4 file4`

file delete: `rm file4, rm *` [rm file কে পুরুষ file delete  
folder name কোরি আসি, folder রিমুভ]

folder delete: `rm dir/` [folder del কোরি থাণ্ড আসি, folder উন্মুক্ত  
কৰে রাখুন]

### Calender

Display calender: `cal`

current month starting from sunday: `cal -3`

system date: `date` <sup>mon</sup> [starting from sunday: `date -S` `date -M`]

month name & month: `date +%m` [month only: `date +%m` [11]]

month name: `date +%b` [month name: `date +%b %m` [nov 11]]

Minitue: `date +%h` [nov]

Day/month: `date +%d` [2:42] → 42

curr port G connected: `who` <sup>am/PM:</sup> `date +%T`

username: `whoami / logname` history, clear

Subject

# Command

Sat	Sun	Mon	Tue	Wed	Thu	Fri
○	○	○	○	○	○	○

Lab-3

sort f2 | nl | cat > f4

Date: / /

Lab-4

arithmetic operation: bc

word count: wc f1

$5+5=10$   
 word count  
 char count  
 Line no  
 2 5 20 f1 file name

only line no: wc -l f1 uc -c f1 wc -w f1

ascending order list:  
increment by 5

- 1. banana
- 2. apple
- 3. guava

multiple command

some: nl f2 | cat > f3

A-Z: sort f2

Z-A: sort -r f2

remove duplicate: sort -u f2

Lab-4

File as 1st to name: head f2  
head -5 f2 (1st 5 to line)

deletion

last 10 to name: tail f2 | tail -5 f2

cut -d " " f5 | cut -f 1,3 f5 field

column wise cut 1 & 3rd column

1-2 [1 to 2]

line wise char cut: cut -c 1-3 f5

file G command copy to column wise / table merge

paste f5 f2

Subject

Lab-5

word | pattern  
file

Sat Sun Mon Tue Wed Thu Fri  
○ ○ ○ ○ ○ ○ ○

Date: / /

search: grep "test" f1 → line 22 there highlight  
→ f1 → case sensitive → insensitive // this is a test  
case insensitive: grep -i "test" f1

count pattern: grep -c "test" f1

grep -i -c "test" f1 // insensitive

only pattern: grep -o "test" f1 // test

Reverse search: 1) line @ pattern(test) तरीके, 2) line तक  
grep -v "test" f1

EST [ EST ] PCS

0

17

1000 0 100 100

1000 0 100 100

4.00

17

EST + EST + PCS = EST, 0 = 0, 100 = 100

(EST + PCS)

EST + PCS + PCS + PCS

17

CS

CamScanner

## Lab-5 : File permission

### Ownership of Linux files

User: file/folder's owner

group: यासिंकना फॉलो ग्रुप एक्सेस करता है।

rwx → execute

4 2 1 → नियंत्रण प्रोग्राम रन -> R W X

User → group → others → value

Command: chmod  $\xrightarrow{\text{user, group, others}} 644$  guest → file/folder

→ change permission

## Final Term

vi editor → for shell programming. [visual editor]

2 mode → 1) command      2) insert mode

by default command mode → for type

file save →  $\text{wq}$  command mode →  $\text{M} \text{H} \text{C}$ , command mode

command → insert [press 'i']  
 ← [esc] command case sensitive.

file create

vi  $f1$  [filename]

save:  $:wq$  [saved and quit]      ;w → save :q → quit

$a$  → cursor up আছে, এখন ; press করলে cursor  
 গুড়ে আছে তখনে ফিল্ট করুন [a i a b c u b]  
 [capital I নিম্ন লাইনে গুড়ে করুন নিম্ন লাইনে হওয়া হবে]

insert mode → নিম্ন লাইনে হওয়া হবে,  
 small 'a' নিম্ন লাইনে হওয়া হবে, এখন মেরে কোথায় কুড়ি হবে,

$a$  →  $a i u a a i b$

capital 'A' → নিম্ন লাইনে কোথায় নিম্ন লাইনে হবে,  
 এখন insert mode → নিম্ন লাইনে কোথায় হবে,

small 'o' → new line create হবে এখন insert mode

capital 'O' → নিম্ন লাইনে কোথায় হবে, এখন insert mode

capital 'U' → নিম্ন লাইনে কোথায় হবে, এখন insert mode

Subject \_\_\_\_\_

Sat Sun Mon Tue Wed Thu Fri  
Date: / /

newline create হলি Give insert mode ক ফিরুজ  
নথি.

virtual keyuse ক উপর ডাউন কেy এর পক্ষে কাজ করে এবং actual  
linux ক কাজ না,

J → ↓ K → ↑ h → ← (left) l → → (right)

backspace / sing char delete; & [press x] → cursor রেখার  
X → cursor রেখার ছাঁচ, অন্য কোনো char delete কর

dd → cursor এর line ক আছে, pure line delete কর

D → cursor এর char ক আছে আছে, & এর char  
ক্ষেত্রে right এর সময় char delete কর কর

[nадим] → [নাম]

line copy : yy → Paste: P → নিচে paste কর, P  
নিচে paste কর

Shift+A → line ক কোথাও কর

3. Bounded waiting: check critical section →  
see waiting list.

### Lab-02

#! /bin/bash → interpreter location [থেকে কোড]

→ ফুল পদ্ধতি [বিবরণ]

Print: echo "Hello World" → কাউন্ট কমান্ড মডেল

Save & exit: Press esc → ;wq

Run      ./filename [chmod 777 f1]

### Shell Variables

- there is no data type for variable নির্ভুল পদ্ধতি  
identify করে নিয়ে।
- variable কখন call/use - কোথা name? ⇒ কার্ড  
\$ use এস্টেশন, \$a.
- numeric first variable name প্রথম ইডেন্টিফার  
⇒ মাঝে space দ্বারা অবচ্ছিন্ন।

Ex: go to insert mode

name=Nadim [no space]

echo "The name is" \$name

Press esc → ;wq

chmod 777 f1

./f1

User input: read name (variable name)

read v1 v2 (run ও স্ট্যাম্প স্পেস থাকলে  
দিয়ে input থাকলে)

newline: /n comment: //

variable  
read -P "Enter the name" name [read -S] → newline  
read -S P → same line

Operators (+ - \* %.)

operator go to next space even mandatory

a = 20 b = 10

sum = \$(expr \$a + \$b)

sum = ~expr \$a + \$b~

(sum = \$(expr \$a + \$b))  
↓ no space

For multiplication: \*

condition: []

Floating point

1) echo "\$num1+\$num2" | bc

→ general space 26.5  
be go to  
next space.

20 :

2) num3 = \$(bc << "\$num1 + \$num2")

read a b  
sum = \$(bc << "\$a + \$b")  
echo \$sum

floating point or multiplication ( ) (57)

2nd n.

Lab-3

# !/bin/bash

read num

if [ \$num -ge 0 ]

then

echo "\$num is positive"

else

echo "\$num is negative."

fi

ending  
of if else  
block

else if

if, elif

switch case

break=;

defamet=\*

Lab-4Loop

B. read num

for (( i=0; i&lt;=num; i++ ))

do

echo \$i

done

1

2

3

4

5

# for number in 1 2 3 4 5

do

echo \$number

done

//ans: 1 2 3 4 5

→ a number is greater  
or equal to 0.

while Loop → reverse an int number

read n

rev=0

while [ \$n -gt 0 ]

do

rem=\$(( n % 10 ))

rev=\$(( rev \* 10 ))

rev=\$(( rev + rem ))

n=\$(( n / 10 ))

done

echo \$rev

for number in {1..10}

do

echo \$number

done

1

2

3

4

5

total

echo = newlineLab-4 → ~~for~~ (O/P trace)

```

1
3
5
7
9
done
for number in {1..10..2}
do
echo $number
done

```

```

for output in $(pwd)
do
echo $output
done
cd mydir

```

Banker's algorithm

Sol) need = max-allocation

$$P_0 = \begin{matrix} 7 & 4 & 3 \end{matrix}$$

$$P_1 = \begin{matrix} 1 & 2 & 2 \end{matrix}$$

$$P_2 = \begin{matrix} 6 & 0 & 0 \end{matrix}$$

$$P_3 = \begin{matrix} 0 & 1 & 1 \end{matrix}$$

$$P_4 = \begin{matrix} 4 & 3 & 1 \end{matrix}$$

safe/unsafe

A = available

AL = Allocation

 $P_0 = \text{cannot}$ 

$$P_1 = \begin{matrix} 1 & 2 & 2 \end{matrix}$$

$$\# = \begin{matrix} 2 & 1 & 0 \end{matrix}$$

P1 গ্রহণ করে যাবে return করে ফিরে, ( $\begin{matrix} 1 & 2 & 2 \\ 2 & 1 & 0 \end{matrix}$ , allocation)

 $\# = \begin{matrix} 5 & 3 & 2 \end{matrix}$ Lab-5 : array $OS = (\text{ubuntu windows kali})$ `echo "${OS[0]}"``echo "${#OS[@]}"` → array size

input:

`read -a array``echo "${array[1]}"`

File parsing

ODE

while read line

do

`echo $line`

done &lt; f1 → line by line parse

IFS = internal field separator (cut)

while IFS '@' read id extra

do

`echo $id`

done &lt; f2

existing check of file

filename = f2

if [ -f \$filename ]

then

`echo "file exist"`else `echo "file not exist"`

fi

need available

↑

↑

Subject

Lab-6

Sat	Sun	Mon	Tue	Wed	Thu	Fri
○	○	○	○	○	○	○

Date: / /

## Function

no return type

#!/bin/bash

```
fact() {
```

```
f=1
```

```
for ((i=1; i<$1; i++))
```

```
do
```

first parameter



```
f=$((f * i))
```

```
done
```

```
echo $f
```

```
g
```

```
result=$(fact $(fact 3))
```

```
echo "factorial = \"$result"
```