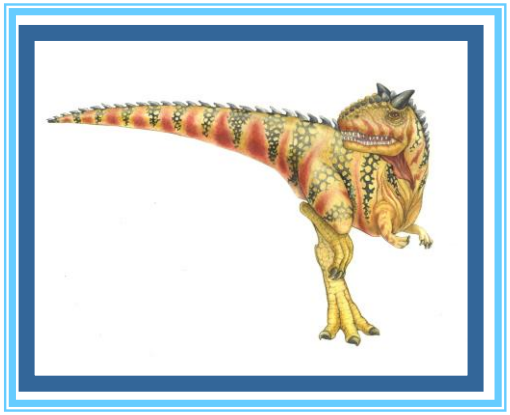
Shell Programming



kiran@jhonson-dell-g3-15: ~/temp1

```
read l
echo "E
read b
((area=l*b))
echo $area
kiran@jhonson-dell-g3-15:~/temp1$
```

Mute Stop Video Security Parti 253 5

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo

Who can see what you share here? Recording On

```
if cond
then

fi
```

Mumbai (J... Desktop

kiran@jhonson-dell-g3-15

Mute | Stop Video | Security | Participants 254 | Chat 5 | New Share | Pause Share | Annotate | Rem

You are screen sharing — Stop Share

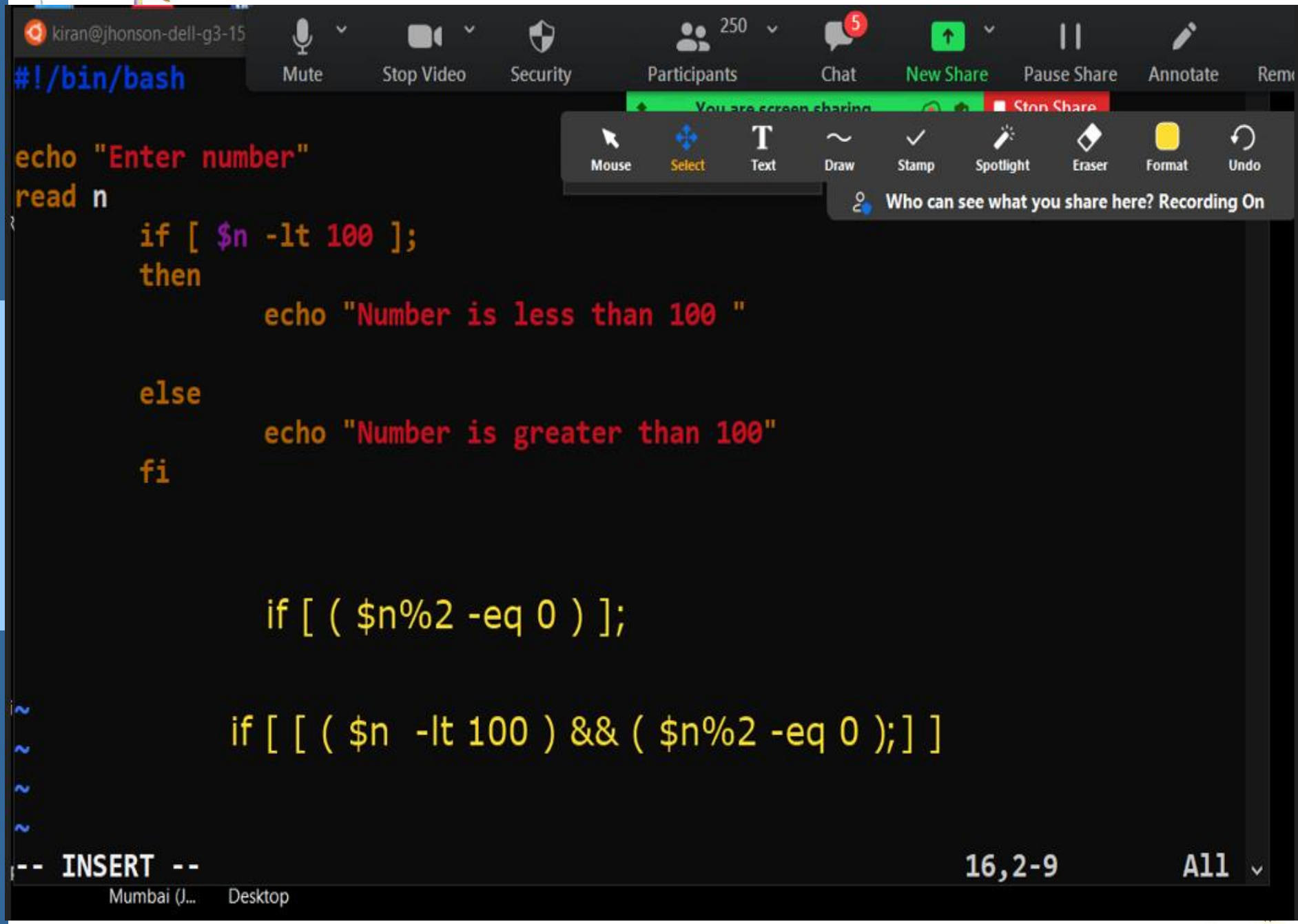
Mouse | Select | Text | Draw | Stamp | Spotlight | Eraser | Format | Undo

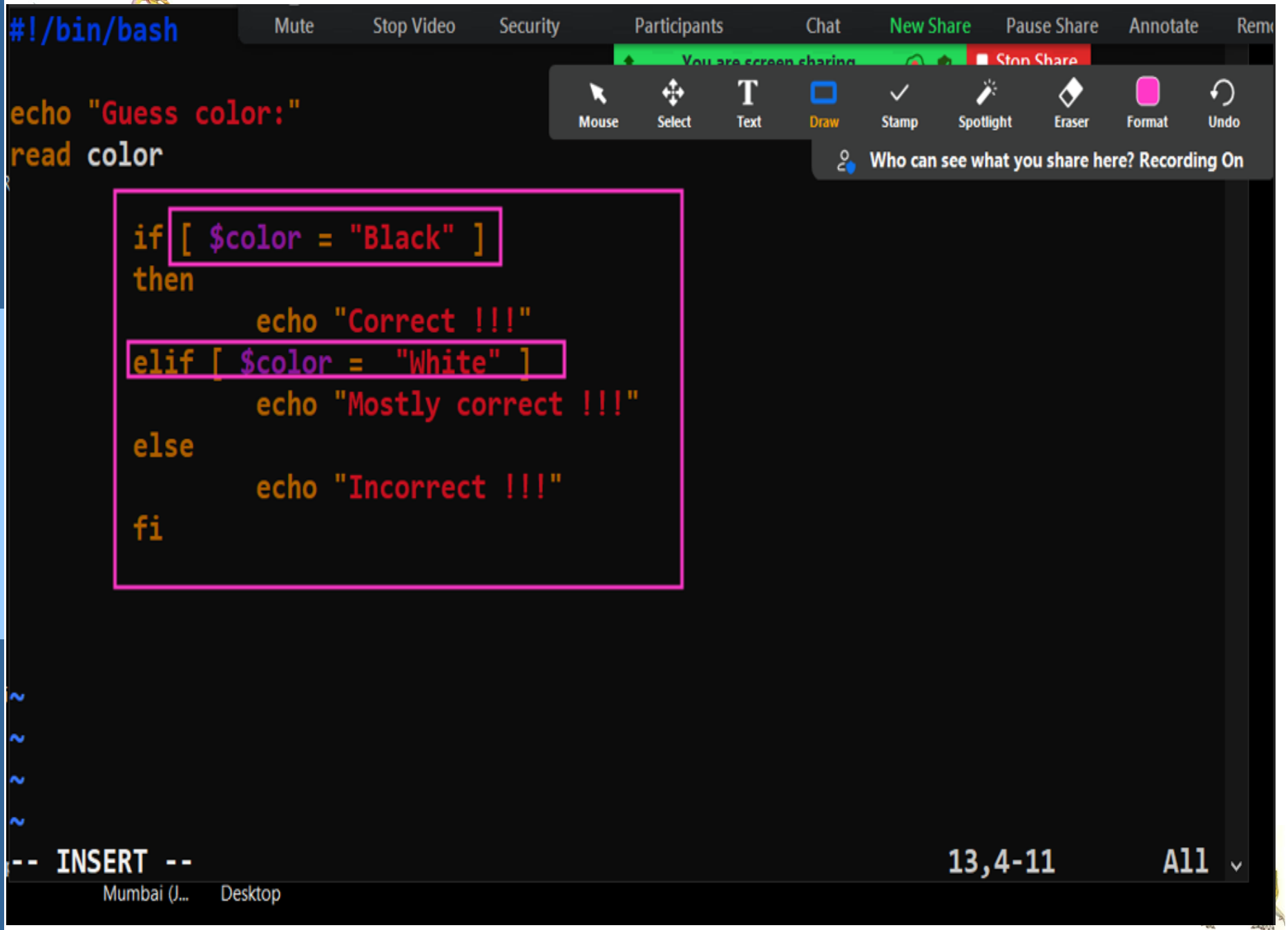
Who can see what you share here? Recording On

```
5
Enter breadth
6
30
kiran@jhonson-dell-g3-15:~/temp1$ cat A5.sh
#!/bin/bash
#calculate area of rectangle.
echo "Enter length:"
read l
echo "Enter breadth"
read b
((area=l*b))
echo $area
kiran@jhonson-dell-g3-15:~/temp1$ vi A6.sh
kiran@jhonson-dell-g3-15:~/temp1$ chmod +x A6.sh
kiran@jhonson-dell-g3-15:~/temp1$ ./A6.sh
Enter color name
pink
Correct!!!
Guess !!!
kiran@jhonson-dell-g3-15:~/temp1$ vi A6.sh
```

Mumbai (J... Desktop

comparison operator


kiran@jhonson-dell-g3-15
Mute
Stop Video
Security
Participants
250
Chat
5
New Share
Pause Share
Annotate
Stop Share
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Who can see what you share here? Recording On
#!/bin/bash
echo "Enter number"
read n
if [$n -lt 100];
then
echo "Number is less than 100 "
else
echo "Number is greater than 100"
fi
if [($n%2 -eq 0)];
if [[($n -lt 100) && ($n%2 -eq 0);]]
-- INSERT --
16,2-9
All
Mumbai (J...
Desktop


#!/bin/bash
Mute
Stop Video
Security
Participants
Chat
New Share
Pause Share
Annotate
Stop Share
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Who can see what you share here? Recording On
echo "Guess color:"
read color
if [$color = "Black"]
then
echo "Correct !!!"
elif [$color = "White"]
echo "Mostly correct !!!"
else
echo "Incorrect !!!"
fi
-- INSERT --
13,4-11
All
Mumbai (J...
Desktop

if statement
if-else statement
if..elif..else..fi statement (Else If ladder)
if..then..else..if..then..fi..fi..(Nested if)
switch statement

if statement
This block will process if specified condition is true.
Syntax:

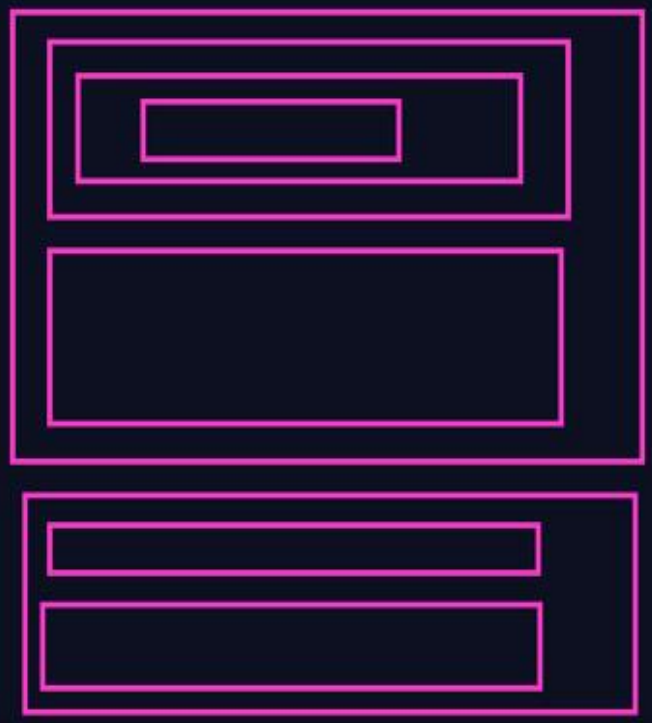
```
if [ expression ]
then
   statement
fi
```

---

if-else statement

Syntax

```
if [ expression ]
then
   statement1
```



---

if..elif..else..fi statement (Else If ladder)

```
if [ expression1 ]
then
   statement1
   statement2
   .
   .
elif [ expression2 ]
then
   statement3
   statement4
   .
   .
else
   statement5
fi
```

```
if [ expression1 ]
then
```

```
   .
else
   statement5
fi

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

if [ expression1 ]
then
   statement1
   statement2
   .
else
   if [ expression2 ]
   then
      statement3
      .
   fi
fi

--------------------------
```


Who can see what you share here? Recording On
#!/bin/bash
i=0
while [$i -lt 10]
do
echo $i
i=`expr $i + 1`
done
-- INSERT --
8,5
All

```
#!/bin/bash
i=0

while [ $i -lt 10 ]
do
        echo $i
        i=`expr $i + 1`
done
```

```
while(cond) → true
{
    .....
}
```

-- INSERT -- 8,5 All

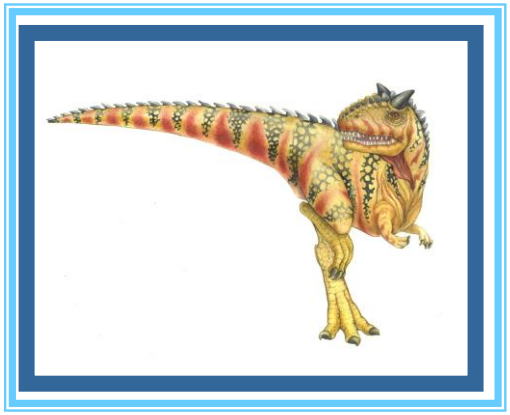
```
#!/bin/bash

Welcome()
{
        echo "My first function."
        echo "$1 $2"
}

Welcome Good morning
~
~
~
~
~
~
~
~
~
"A12.sh" 9L, 98C                                    6,19          All
```

Processes
Day3: Sep 2021

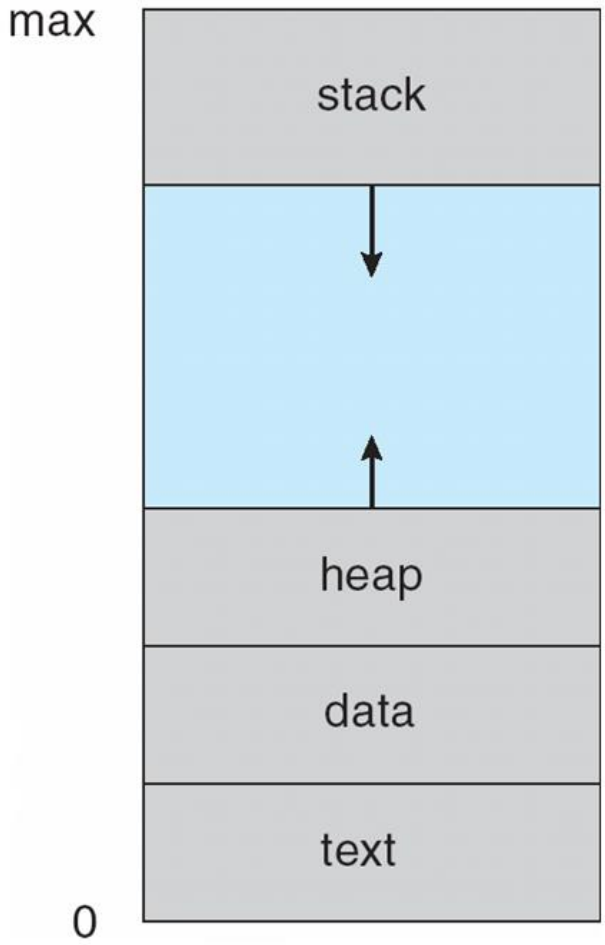
Kiran Waghmare



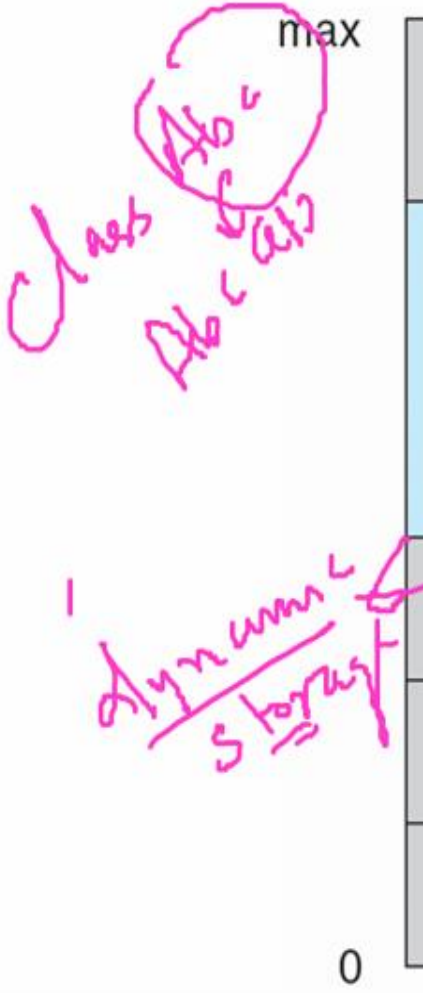
Agenda: Processes

- Preemptive and non preemptive
- Process mgmt
- Process life cycle
- Schedulers
- Scheduling algorithms
- Creation of fork, waitpid, exec system calls
- Orphan and zombie

Process in Memory



Process in Memory



Process Concept

- An operating system executes a variety of programs:
  - Batch system – jobs
  - Time-shared systems – user programs or tasks
- Textbook uses the terms *job* and *process* almost interchangeably
- **Process – a program in execution; process execution must progress in sequential fashion**
- A process includes:
  - program counter
  - stack
  - data section

Process State

- As a process executes, it changes *state*
  - **new**: The process is being created
  - **running**: Instructions are being executed
  - **waiting**: The process is waiting for some event to occur
  - **ready**: The process is waiting to be assigned to a processor
  - **terminated**: The process has finished execution

Diagram of Process State

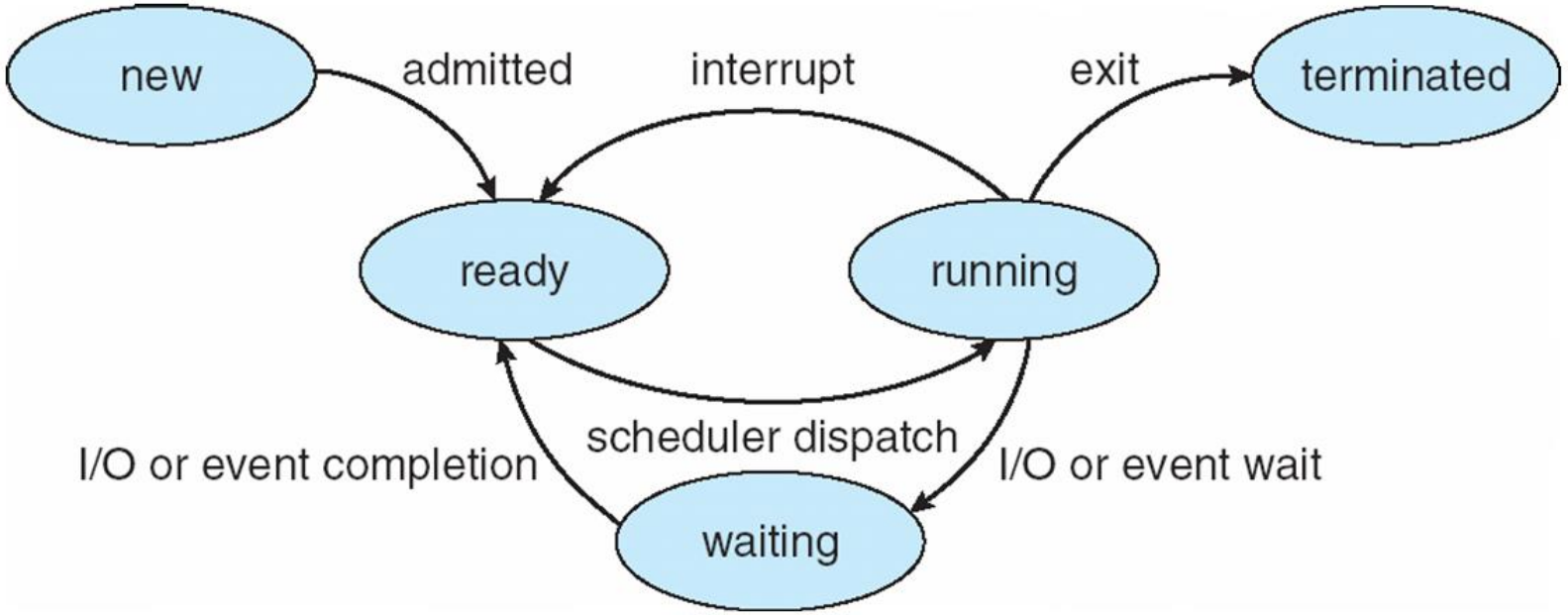
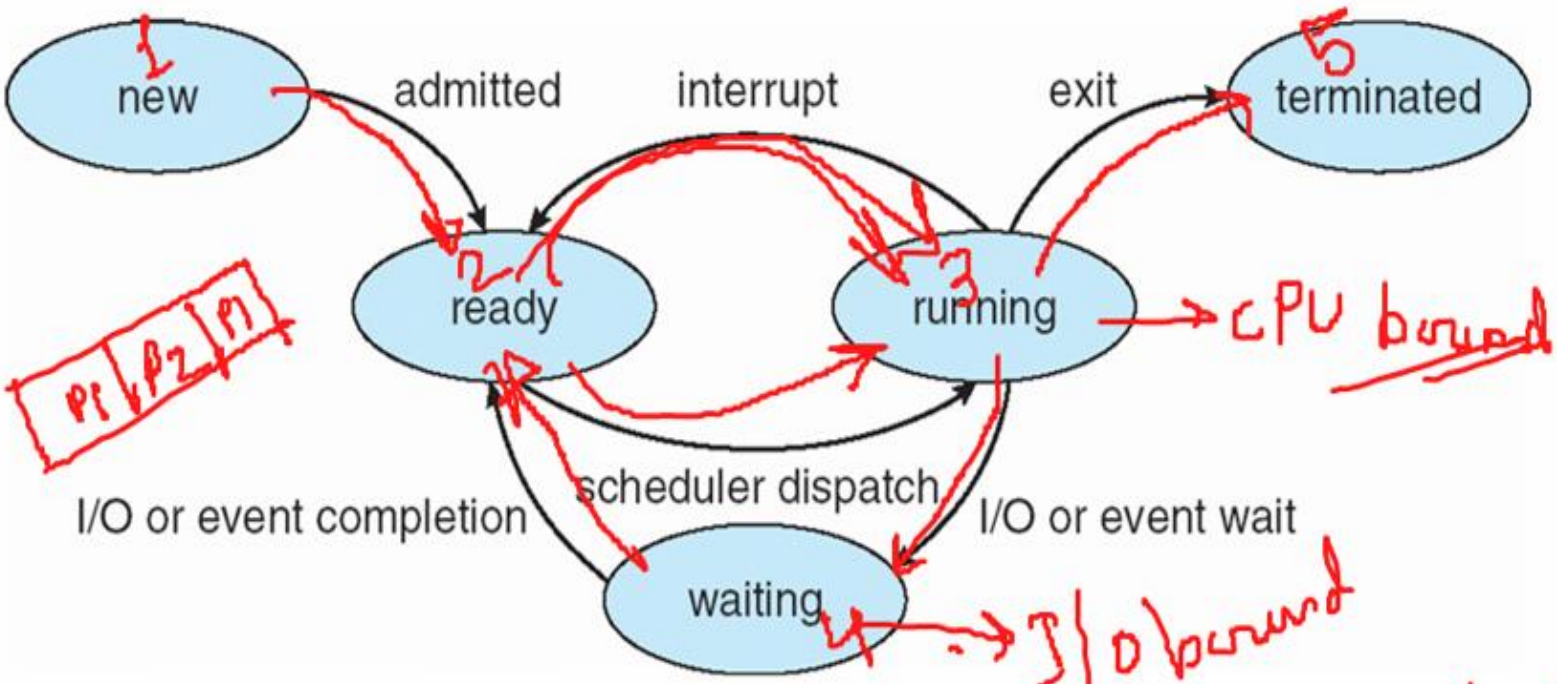


Diagram of Process State



Process Control Block (PCB)

Information associated with each process

- Process state
- Program counter
- CPU registers
- CPU scheduling information
- Memory-management information
- Accounting information
- I/O status information

Process Control Block (PCB)