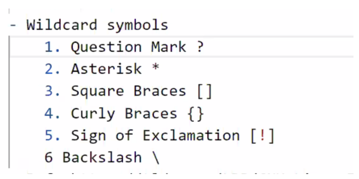
5️⃣ DAY 5

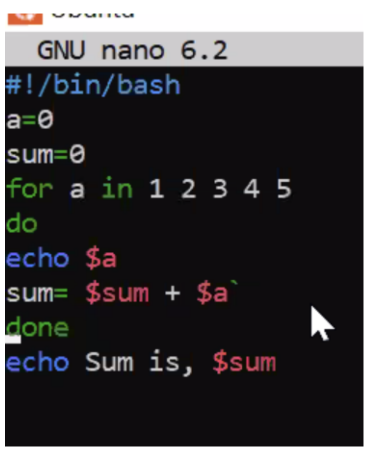
💡 Date : 31 Aug 2024

- In shell all of these for , echo is a cmd not programming . It's a script not a programming lang .
- Shell can be accessed by users using a command line interface. A special program called Terminal in Linux/macOS, or Command Prompt in Windows OS is provided to type in the human-readable commands such as "cat", "ls" etc.
- **IMP ⇒ In for we need not to write the increment but in while we need to write the increment statement .**

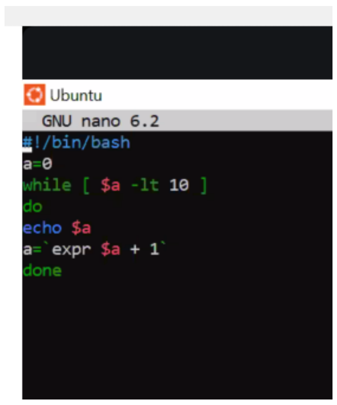
```
   Loops : It is used to repeat certain code upto n number of time .
   => 1 .
   Syntax : a is the counter variable
   a=0
   for a in  1 2 3 4 5  //these are the value we want to start iterat
   // Each time the for loop executes, the value of the variable var
   list of words, word1 to wordN.
   do

   echo hello
   echo $a
   done
   =>
   To run this => bash p1 (Name)


   2. sum of first n numbers =>
   sum=0
   a = 0
   for a in  1 2 3 4 5  //these are the value we want to start iterat
   do
   echo $a  w
   sum = `expr $sum + $a` // redefining the var , to make terminal ur
     (backticks) and expr keyword followed by space
   done
   echo sum is $sum

   3. While Loop : condiiton is given inside []
   => Syntax
   while [condition]
   do
   statement
   increment
   done

   => nano file_name
   a=0
   while[$a -lt 10] // less than 10  run uptil 9 (0-9)
   do
   echo $a
// to increment the value of a by 1
   a=`expr $a + 1`
   done

   bash file_name
```



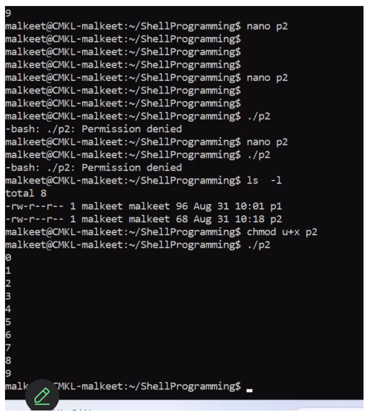
```
- Wildcard symbols
    1. Question Mark ?
    2. Asterisk *
    3. Square Braces []
    4. Curly Braces {}
    5. Sign of Exclamation [!]
    6 Backslash \
```



```
  GNU nano 6.2
#!/bin/bash
a=0
sum=0
for a in 1 2 3 4 5
do
echo $a
sum= $sum + $a`
done
echo Sum is, $sum
```

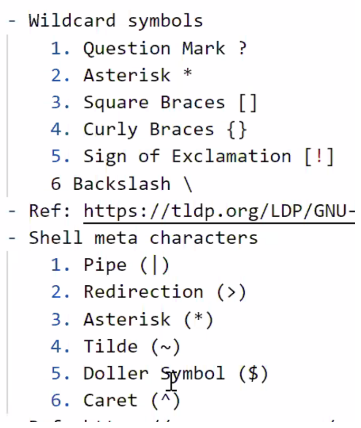


```
Ubuntu
  GNU nano 6.2
#!/bin/bash
a=0
while [ $a -lt 10 ]
do
echo $a
a=`expr $a + 1`
done
```

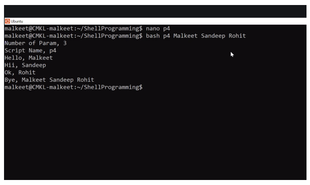


```
9
malkeet@CMKL-malkeet:~/ShellProgramming$ nano p2
malkeet@CMKL-malkeet:~/ShellProgramming$
malkeet@CMKL-malkeet:~/ShellProgramming$
malkeet@CMKL-malkeet:~/ShellProgramming$
malkeet@CMKL-malkeet:~/ShellProgramming$ nano p2
malkeet@CMKL-malkeet:~/ShellProgramming$
malkeet@CMKL-malkeet:~/ShellProgramming$
malkeet@CMKL-malkeet:~/ShellProgramming$ ./p2
-bash: ./p2: Permission denied
malkeet@CMKL-malkeet:~/ShellProgramming$ nano p2
malkeet@CMKL-malkeet:~/ShellProgramming$ ./p2
-bash: ./p2: Permission denied
malkeet@CMKL-malkeet:~/ShellProgramming$ ls  -l
total 8
-rw-r--r-- 1 malkeet malkeet 96 Aug 31 10:01 p1
-rw-r--r-- 1 malkeet malkeet 68 Aug 31 10:18 p2
malkeet@CMKL-malkeet:~/ShellProgramming$ chmod u+x p2
malkeet@CMKL-malkeet:~/ShellProgramming$ ./p2
0
1
2
3
4
5
6
7
8
9
malk  CMKL-malkeet:~/ShellProgramming$
```

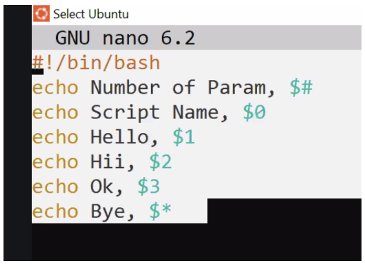
```
    4. We can execute shell script like "./file_name "


    5. Until Loop =>
    a=0
    until [$a -gt 10] // until a is greater than 10 ,  run until 10 .
     (0-10) => jabtak a greater than 10 na ho jaye tab tak chalana
    do
    echo $a
    a=`expr $a + 1`
    done

    => bash file_name

    6. What are Wildcard symbol ?
    Ans .
    They can be used anywhere with any set of cmd => Case - sensitive
     if i want to search and print that lines inside my file with part
     and ending with some words.


    When we have multiple touch files1 file2 file3 file4 set1 set2 set
    i want to search some files with respect to given parameters ,
    cat fil*  (particular word )

    7. if i want to search and print that lines inside my file with pa
    with some words.

    nano file1
    hi fefowfjef
    h dkskadk mdkwqdskws
    newewekwkewdkmq
    hello djsdjsjd sdnsmagyqywndjz


    cat file1 // print all the lines
    cat file1 |
    grep '^h.*n$' file1

    grep: This is the command-line tool we use for searching patterns
  ^h: This part of the pattern says "the line must start with the char
  The ^ signifies the start of a line.
  .*n: This says "followed by any number of any characters (. matches
   zero or more occurrences), and ending with the character 'n'".
   $: This anchors the pattern to the end of the line, ensuring that '
   file1: This specifies the file you want to search in.



8 . (~ cd) used for going back to home directory or simply using cd .
9. Search how list all the file starting with s or certain word .
10. backlash is used to move between directories and sometimes to prir



11. Command Line Arguements => The data given during the cmd executing
nano p1
echo hello
echo hi
echo okay
echo bye
bash p1

    while running the p1 file i want to give some data with the file w
    "bash p1 Arg1 Arg2 Arg3 Arg4 "
```



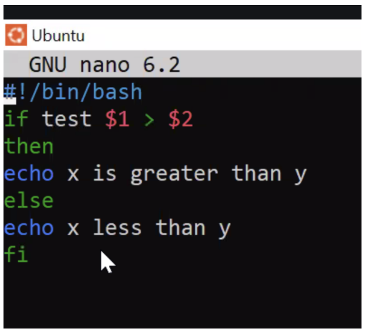
- Wildcard symbols
    1. Question Mark ?
    2. Asterisk *
    3. Square Braces []
    4. Curly Braces {}
    5. Sign of Exclamation [!]
    6 Backslash \
- Ref: https://tldp.org/LDP/GNU-
- Shell meta characters
    1. Pipe (|)
    2. Redirection (>)
    3. Asterisk (*)
    4. Tilde (~)
    5. Doller Symbol ($)
    6. Caret (^)



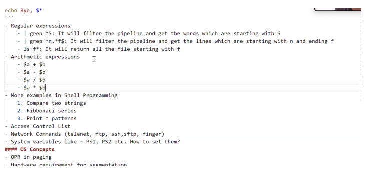
```
malkeet@CMKL-malkeet:~/ShellProgramming$ nano p4
malkeet@CMKL-malkeet:~/ShellProgramming$ bash p4 Malkeet Sandeep Rohit
Number of Param, 3
Script Name, p4
Hello, Malkeet
Hii, Sandeep
Ok, Rohit
Bye, Malkeet Sandeep Rohit
malkeet@CMKL-malkeet:~/ShellProgramming$
```



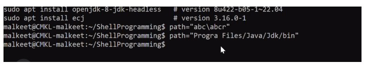
```
GNU nano 6.2
#!/bin/bash
echo Number of Param, $#
echo Script Name, $0
echo Hello, $1
echo Hii, $2
echo Ok, $3
echo Bye, $*
```



```
GNU nano 6.2
#!/bin/bash
if test $1 > $2
then
echo x is greater than y
else
echo x less than y
fi
```



```
echo Bye, $*
```
- Regular expressions
    - | grep ^S: Tt will filter the pipeline and get the words which are starting with S
    - | grep ^n.*f$: It will filter the pipeline and get the lines which are starting with n and ending f
    - ls f*: It will return all the file starting with f
- Arithmetic expressions
    - $a + $b
    - $a - $b
    - $a / $b
    - $a * $b
- More examples in Shell Programming
    1. Compare two strings
    2. Fibbonaci series
    3. Print * patterns
- Access Control List
- Network Commands (telenet, ftp, ssh,sftp, finger)
- System variables like - PS1, PS2 etc. How to set them?
OS Concepts
- OPR in paging
- Hardware requirement for segmentation



```
sudo apt install openjdk-8-jdk-headless   # version 8u422-b05-1~22.04
sudo apt install ecj                      # version 3.16.0-1
malkeet@CMKL-malkeet:~/ShellProgramming$ path="abc\abcr"
malkeet@CMKL-malkeet:~/ShellProgramming$ path="Progra Files/Java/Jdk/bin"
malkeet@CMKL-malkeet:~/ShellProgramming$
```

```
    inside the p1 file we will write
    echo number of para, $#
    echo script name, $0
    echo hello, $1 // will print first parameter
    echo hello, $3 // print 3rd parameter
    echo bye , $* // print all the parameter



    12.
    x=100
    y=100
    if test [$x -eq $y ]
    then
    echo x an y are equal
    else
    echo x and y are not equal
    fi


    x=Aditi
    y=1000
    if test $x -eq $y
    then
    echo x an y are equal
    else
    echo x and y are not equal
    fi

  For Strings =>
    x=Aditi
    y=Aditi
    if test [$x == $y ]
    then
    echo x an y are equal
    else
    echo x and y are not equal
    fi


    "Both words are equal => in uniq => it will return 0 , means true


    13 .
    x=Aditi
    y=Mehre
    if test $x > $y // Comparing two strings.
    then
    echo x is greater than y
    else
    echo y is greater than x
    fi


    14 . test -f file_name +> file
         test -d => to check directory is present or not
         test -s file_name => to check file is emoty or not


15. Write a script which will accept two words from cmd and compare th

    x=Aditi
    y=Mehre
    if test $1 > $2 // Comparing two strings.
    then
```

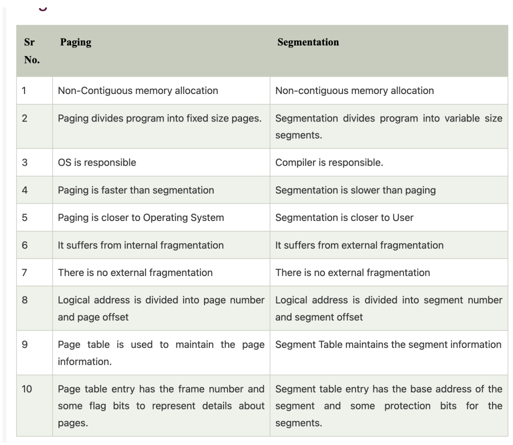
```
   echo x is greater than y
   else
   echo y is greater than x
   fi
   and pass the cmd arguments => bash file_name  "Aditi" "Mehre"


  16. echo enter a number
       read num1
       echo enter a  number
       read num2
         res = `expr $num1 + $num2`
           echo result , $res

     This will work with +,-,/ only not with * because it is a wildca

     echo enter a number
       read num1
       echo enter a  number
       read num2
         res = `expr $num1 /* $num2`
           echo result , $res

           For * we have to use backslash(escape char ) with it .

   17 . To set path we use Path cmd
   18. Kill cmd => to terminate the process .
```

OS Concepts :

1. Multi Level Paging or Two level Paging
2. Multi Level Paging ⇒ Larger the Virtual memory larger the Page table to get stored in frames , Then we have to divide the page table to further pages .
3. Two Level Paging ⇒

⇒ Listen to Recording

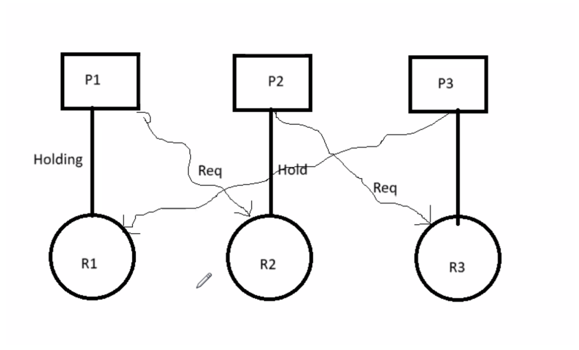
1. OPR Optimal Page Replacement Algorithm
2. Hardware Requirement for Segmentation :
    a. Virtual + Physical + Cache
3. Difference between Segmentation and Paging



Sr No.	Paging	Segmentation
1	Non-Contiguous memory allocation	Non-contiguous memory allocation
2	Paging divides program into fixed size pages.	Segmentation divides program into variable size segments.
3	OS is responsible	Compiler is responsible.
4	Paging is faster than segmentation	Segmentation is slower than paging
5	Paging is closer to Operating System	Segmentation is closer to User
6	It suffers from internal fragmentation	It suffers from external fragmentation
7	There is no external fragmentation	There is no external fragmentation
8	Logical address is divided into page number and page offset	Logical address is divided into segment number and segment offset
9	Page table is used to maintain the page information.	Segment Table maintains the segment information
10	Page table entry has the frame number and some flag bits to represent details about pages.	Segment table entry has the base address of the segment and some protection bits for the segments.

DeadLock Working ⇒

1. We have multiple processes and we have some resources , it can mean memory and address or any device .
2. Process P1 is working with R1 resource asking for R2 , P2 is using R2 asking for R3 , P3 is holding R3 but asking for R1



3. OS will be not be completing the execution because each process is holding some resources and not letting go the current resources and none is getting the resources and hence it result in circular wait and cause a deadlock means a traffic jam and locked position where no one is doing anything rather cause a locked condition

How it can be happen : They all should occur together

1. Mutual exclusion ⇒
   a. Any process can request request a resources but holding process can only give the permission .
   b. Resources can share resources but one process should have at least one process.
2. No preemption ⇒ no one can preempt it .
3. Hold and Wait ⇒ Process are allowed to make request for other resources while holding one .
4. Circular Wait ⇒

DeadLock Handling Techniques :

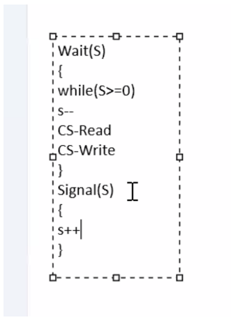
1. **Deadlock avoidance or Ignorance** :
   a. If a deadlock is very rare, then let it happen and reboot the system. This is the approach that both Windows and UNIX take. we use the ostrich algorithm for deadlock ignorance.
   b. In Deadlock, ignorance performance is better than the above two methods but the correctness of data is not there.
2. **Deadlock Prevention :**
   a. No Mutual exclusion :
   b. Make a preemption system
   c. No Hold and wait
   d. No circular wait
3. **Deadlock Recovery**
   a. Kill a Process
   b. Add multiple instances of resources .
   c. Resource allocation Graph .

PROCESS SYNCHRONIZATION : (YOUTUBE + GFG) Imp in CCEE

If bank is not process synchronized , bank have to be synchronized means for a account process should be allowed to enter one by one not by all at once . Allowing the process to get any resources one by one is known as **process synchronization in critical section(Sensitive memory where at a time only one person can have access) .** They are managed by the following :

⇒ Semaphore (variable) and Mutex (Object)

1. If P1 want to enter into critical section ⇒ it will pass the mutex object as a system call ⇒ wait () . Mutex is actually is a Lock and after the resources is free , signal system is generated and it is unlocked . and the other process will be allowed to enter.
2. **Semaphore** (s): To manage multiple process and synchronize with similar instance , it allow everyone to read it , but for writing one by one . (1 , 0 , -1)

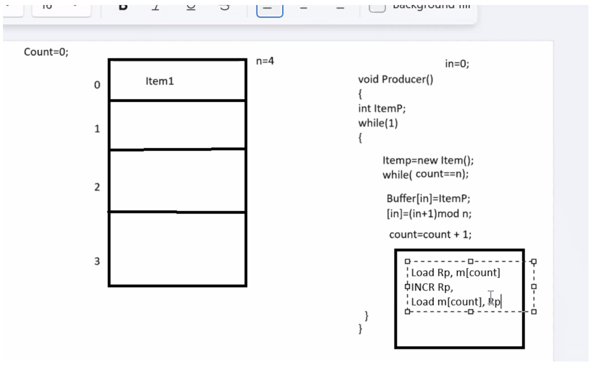


```
Wait(S)
{
while(S>=0)
S--
CS-Read
CS-Write
}
Signal(S)
{
S++
}
```

DEADLOCK VS STARVATION ⇒ Starvation is bounded wait and deadlock is not bounded wait .

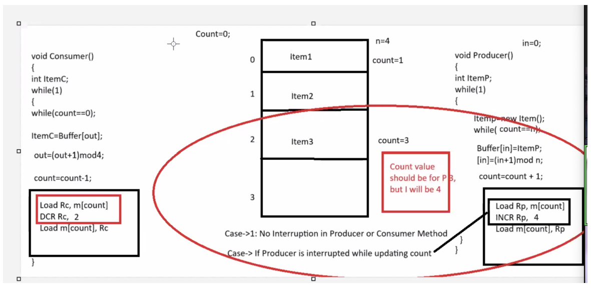
Producer Consumer Problem

1. Job or Producer is to produce item (Programs , etc) and will keep in stack, heap , data etc .
2. Semaphore and mutex is the solution for this .



```
Count=0;
in=0;
void Producer()
{
int ItemP;
while(1)
{
    Itemp=new Item();
    while( count==n);

    Buffer[in]=ItemP;
    [in]=(in+1)mod n;

    count=count + 1;
}
}
```

```
Load Rp, m[count]
INCR Rp,
Load m[count], Rp
```



```
Count=0;
void Consumer()
{
int ItemC;
while(1)
{
while(count==0);

ItemC=Buffer[out];

 out=(out+1)mod4;

 count=count-1;
```

```
Load Rc, m[count]
DCR Rc,  2
Load m[count], Rc
}
```

```
in=0;
void Producer()
{
int ItemP;
while(1)
{
    Itemp=new Item();
    while( count==n);

    Buffer[in]=ItemP;
    [in]=(in+1)mod n;

    count=count + 1;
}
}
```

```
Load Rp, m[count]
INCR Rp,  4
Load m[count], Rp
```

Count value should be for P 3, but I will be 4

Case->1: No Interruption in Producer or Consumer Method

Case-> If Producer is interrupted while updating count

Topic: Recording
Module: COS
Date: 31/08/24
Session: Morning

https://us02web.zoom.us/rec/share/XaKCizqlCPou40NUa8w_1p0FA_tuwn6ijBQf59e7raDqgLROM0rfVeC_tNdRkio1.C_kj03g9NTdGBbm2

Topic: Recording
Module: COS
Date: 31/08/24
Session: Afternoon

https://us02web.zoom.us/rec/share/-6NnD6Ev--eCDYXBWxw5khHjC6dhlH_qcvwnSDVr52JWI4D_sAM8vqNqgzB4Ga6Z.fvr3SvvMYZ-IqPvs