SYLLABUS

Pr.1-OPERATINGSYSTEMLAB

Total Periods	**60**	**MaximumMarks**	**50Marks**
Lab. Periods:	**4Periods/week**	**TermWorks**	**25Marks**
Examination	**3hours**	**EndSemesterExamination**	**25Marks**

A.LISTOFPRACTICALS:-

1. WriteaShellscripttoprintthecommandlineargumentsinreverseorder.
2. WriteaShellscripttocheckwhetherthegivennumberispalindromeornot.
3. WriteaShellscripttosortthegivenarrayelementsinascendingorderusingbubble sort.
4. WriteaShellscripttoperformsequentialsearchonagivenarrayelements.
5. WriteaShellscripttoperformbinarysearchonagivenarrayelements.
6. Writea Shellscripttoacceptanytwofilenamesandchecktheirfilepermissions.
7. Write a Shell script to read a path name, create each element in that path e.g: a/b/c i.e.,'a'isdirectoryin thecurrentworkingdirectory,under 'a'create'b',under'b'create 'c'.
8. WriteaShellscripttoillustratethecase-statement.
9. WriteaShellscripttoacceptthefilenameasargumentsandcreateanothershell script, which recreates these files with its original contents.
10. WriteaShellscripttodemonstrateTerminallocking.
11. Writea Shellscripttoacceptthevalid loginname,ifthe loginnameisvalidthenprint its home directory else an appropriate message.
12. WriteaShellscripttoreadafilenameandchangetheexistingfilepermissions.
13. WriteaShellscripttoprintcurrentmonthcalendarandtoreplacethecurrentday number by '*' or '**' respectively.
14. Writea ShellScripttodisplaya menuconsistingof optionstodisplaydiskspace,the current users logged in, total memory usage, etc. (using functions.)
15. WriteaC-programtoforkachildprocessandexecutethegivenLinuxcommands.
16. Writea C-programtoforkachildprocess, printownerprocessIDanditsparent process ID.
17. Writea C-programtoprompt theuserforthe name of theenvironmentvariable,check its validity and print an appropriate message.
18. WriteaC-programtoREADdetailsof Nstudentssuchasstudentname,regnumber, semester and age. Find the eldest of them and display his details.

BooksRecommended:-

Sl.No	Nameof Authors	TitleoftheBook	Nameofthe publisher
1	SumitabhaDas,4thEdition,	"UNIX–Concepts andApplications",	TataMcGrawHill, 2006.
3	YashvantKanetkar	UnixSellProgramming 1st edition	BPBPublication

WhatisOS?

The operating system helps in improving the computer software as well as hardware.Without OS, it became very difficult for any application to be user-friendly. The Operating System provides a user with an interface that makes any application attractive and user- friendly. The operating System comes with a large number of device drivers that make OS services reachable to the hardware environment. Each and every application present in the system requires the Operating System. The operating system works as a communication channel between system hardware and system software. The operating system helps an application with the hardware part without knowing about the actual hardware configuration. It is one of the most important parts of the system and hence it is present in every device, whether large or small device.

FunctionofOS?

- MemoryManagement
- ProcessorManagement
- DeviceManagement
- File Management
- Security
- Controloversystemperformance
- Job accounting
- Errordetecting aids
- Coordinationbetweenothersoftwareandusers
-

Whatis Kernel?

A kernel is the core part of an operating system. It acts as a bridge between software applications and the hardware of a computer. The kernel manages system resources, such as the CPU, memory, and devices, ensuring everything works together smoothly and efficiently.Ithandlestaskslikerunningprograms,accessingfiles,andconnectingto devices like printers and keyboards.

WhatisMicrokernel?

Microkernel is one of the classification of the kernel. Being a kernel it manages all system resources.Butinamicrokernel,the**user services**and**kernelservices**areimplementedin different address space. The user services are kept in **user address space**, and kernel

servicesarekeptunder **kerneladdressspace**,thusalsoreducesthesizeofkernel andsize of operating system as well.

TypesofOperatingSystems

TherearesevеraltypesofOperatingSystemswhicharementionedbelow.

- BatchOperatingSystem
- Multi-ProgrammingSystem
- Multi-ProcessingSystem
- Multi-TaskingOperatingSystem
- Time-SharingOperatingSystem
- DistributedOperatingSystem
- NetworkOperatingSystem
- Real-TimeOperatingSystem

PROCESSSCHEDULING

- The process scheduling is the activity of the process manager that handles the removalof the runningprocessfromtheCPUandtheselectionofanotherprocesson thebasisofaparticular strategy**.**
- Process scheduling is an essential part of a Multiprogramming operating systems. Such operatingsystemsallowmorethanoneprocesstobeloadedintotheexecutablememoryata time and the loaded process shares the CPU using time multiplexing

FirstComeFirstServe(FCFS)

- Jobsareexecutedonfirstcome,firstservebasis.
- Itisanon-preemptive,pre-emptiveschedulingalgorithm.
- Easytounderstandandimplement.
- Itsimplementationisbased onFIFOqueue.
- Poorinperformanceasaveragewaittime ishigh.

WhatisProcess Management?

Process management involves various tasks like creation, scheduling, termination of processes, and a dead lock. Process is a program that is under execution, which is an important part of modern-day operatingsystems.The OSmustallocateresources thatenableprocessesto shareand exchange information.Italsoprotectsthresourcesofeachprocessfromothermethods andallowssynchronization among processes.

ProcessControlBlocks(PCB)

The PCBisafull formofProcessControlBlock.Itisadata structure thatismaintainedbytheOperating System for every process. The PCB should be identified by an integer Process ID

(PID).Ithelpsyoutostorealltheinformationrequiredtokeeptrackofalltherunning processes

INTERPROCESSCOMMUNICATION(IPC)

IndependentProcessesoperatingconcurrentlyonasystemsarethosethatcanneither affect otherprocesses or be affected by other processes.

Cooperatingprocessesarethosethatcanaffectorareaffectedbyother**processes** runningon thesystem.**Cooperatingprocesses**maysharedatawitheachother

SWAPPING

Swapping isa mechanismin which a processcan be swapped temporarily outof main memory(ormove)tosecondarystorage(disk)andmakethatmemoryavailabletoother processes.Atsomelater time, the systemswaps back the process fromthe secondary storage to main memory.

PAGEFAULT

Apagefaulthappenswhenarunningprogramaccessesamemorypagethatismappedinto the virtualaddress space, but not loaded in physical memory.

Sinceactualphysicalmemoryismuchsmallerthanvirtualmemory,pagefaultshappen. Incaseofpagefault,OperatingSystemmighthavetoreplaceone oftheexistingpages with the newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace.The target for all algorithms is to reduce the number of page faults

.

Experiments 1:

1. WriteaShellscripttoprintthecommandlineargumentsinreverseorder.

```
#!/bin/bash
#Scripttoprintcommandlineargumentsinreverseorder Echo
“Arguments in reverse order:”
For((i=$#;i>=1;i--));do Echo
  “${!i}”
Done
```

Experiments 2:

2. WriteaShellscripttocheckwhetherthegivennumberispalindromeor not.

```
#!/bin/bash
#Scripttocheckifanumberisapalindrome Echo
“Enter a number:”
Readnum
Reverse=$(echo“$num”|rev)
If[“$num”-eq“$reverse”];then
Echo “$num is a palindrome”
Else
  Echo“$numisnotapalindrome” Fi
```

Experiments 3:

3. WriteaShellscripttosortthegivenarrayelementsinascendingorderusing bubble sort.

```
#!/bin/bash
#Scripttosortarrayusingbubblesort
Echo“Enterthearrayelementsseparatedbyspace:” Read -a
arr

    N=${#arr[@]}

For((i=0;i<n-1;i++));do
  For((j=0;j<n-i-1;j++));do
    If[“${arr[j]}”-gt“${arr[j+1]}”];then Temp=${arr[j]}
      Arr[j]=${arr[j+1]}
    Arr[j+1]=$tempFi
  Done
Done

Echo“SortedArray: ${arr[@]}”
```

Experiments 4:

4. WriteaShellscripttoperform sequentialsearchonagivenarrayelements.

```
#!/bin/bash
#Scriptto performsequentialsearch
Echo“Enterarrayelementsseparatedbyspace:” Read
-a arr
Echo“Entertheelementtosearch:”
Read key
Found=0
Fori in“${arr[@]}”;do
  If[“$i”-eq“$key”];then
    Found=1
    Break
  Fi

Done

If[$found-eq 1 ];then
  Echo“$keyfoundinthearray”
Else
  Echo“$keynotfoundinthearray” Fi
```

Experiments 5:

5. WriteaShellscripttoperformbinarysearchonagivenarrayelements.

```
#!/bin/bash
#Scripttoperformbinarysearch Binary_search() {
  Localarray=(“$@”)
  Local low=0
  Local high=$((${#array[@]} – 1))
  Echo“Entertheelementtosearch:” Read
  key
  While[$low-le$high];do
    Mid=$(((low + high) / 2))
    If [ “${array[mid]}” -eq “$key” ]; then
      Echo“$keyfoundatposition$mid” Return
    Elif[“${array[mid]}”-lt“$key”];then Low=$((mid +
      1))
    Else
      High=$((mid–1)) Fi

  Done
  Echo“$keynot found”
}

Echo“Entersortedarrayelementsseparatedbyspace:” Read -a
arr
Binary_search“${arr[@]}”
```

Experiments 6:

6. WriteaShellscripttoacceptanytwofilenamesandchecktheirfile permissions.

```
#!/bin/bash

#Scripttocheckfilepermissions

Echo “Enter the first file name:”

Read file1

Echo“Enterthesecondfilename:”

Read file2


Forfilein“$file1”“$file2”;do If [
  -f “$file” ]; then
    Echo“Permissionsfor$file:” Ls
  -l “$file” | awk ‘{print $1}’ Else
    Echo“$filedoesnotexist.” Fi
Done
```

Experiments 6:

7. WriteaShellscriptto read apathname,createeachelementinthatpath

```
#!/bin/bash
#Scripttocreatedirectoriesfromapath echo
"Enter a path (e.g., a/b/c):"
readpath
IFS='/'read-ra dirs<<<"$path"

fordirin"${dirs[@]}";do mkdir
  -p "$dir"
  cd"$dir"
done

echo"Path$pathcreated."
```

Experiments 8:

8. WriteaShellscripttoillustratethecase-statement.

```
#!/bin/bash
#Scripttodemonstratecasestatement Echo
“Enter a number (1-3):”
Read num
Case$numin
  1) echo“Youselectedoption1”;;
  2) echo“Youselectedoption2”;;
  3) echo“Youselectedoption3”;;
  *)echo“Invalidoption”;;
Esac
```

Experiments 9:

9. WriteaShellscripttocreatefileswithitsoriginal contents.

```
#!/bin/bash

#Scripttorecreatefiles

Echo“Enterfilenamesseparatedbyspace:” Read

-a files

Forfilein“${files[@]}”;do If [

-f “$file” ]; then

    Cp“$file” “$file.bak”

    Echo“Recreated$fileas$file.bak”

  Else

    Echo“$filedoesnotexist.” Fi

Done
```

Experiments 10:

10. WriteaShellscripttodemonstrateTerminallocking.

```
#!/bin/bash
#Script tolock the terminal
Echo“Enterthepasswordtolockthe terminal:”
Read -s password
Echo“Terminallocked.Enterthepasswordtounlock:” While
:; do
  Read -s try
  If[“$try”==“$password”];then
    Echo “Terminal unlocked.”
    Break
  Else
    Echo“Wrongpassword.Tryagain.” Fi

Done
```

Experiments 11:

11. WriteaShellscripttoacceptthevalidloginname,iftheloginnameisvalid then print its home directory else an appropriate message.

```
#!/bin/bash

#Scripttodisplaythehomedirectoryofavalidloginname Echo

“Enter a login name:”

Readlogin

If id “$login” &>/dev/null; then

  Home_dir=$(evalecho“~$login”)

  Echo“Homedirectoryof$loginis:$home_dir”

Else

  Echo“Invalidloginname.” Fi
```

Experiments 12:

12. WriteaShellscripttoreadafilenameandchangetheexistingfile permissions.

```
#!/bin/bash

#Scripttochangefilepermissions

Echo “Enter the file name:”

Readfile

If[ -f “$file” ]; then

  Echo“Enternewpermissions(e.g.,755):” Read

  perms

  Chmod“$perms”“$file”

  Echo“Permissionsof$filechangedto$perms.” Else

  Echo“$filedoesnotexist.”
  Fi
```

Experiments 13:

13. WriteaShellscripttoprintcurrentmonthcalendarandtoreplacethecurrent day number by '*' or '' respectively.**

```
#!/bin/bash

#Scripttodisplaythecalendarwiththecurrentdayreplaced Cal |

sed "s/\b$(date +%d)\b/**/"
```

Experiments 14:

14. WriteaShellScriptto displayamenuconsisting ofoptionstodisplaydisk space,thecurrentusersloggedin,totalmemoryusage,etc.(usingfunctions.)

```
#!/bin/bash
#Scripttodisplayamenuforsystemresources
While true; do
  Echo“Menu:”
  Echo “1. Display disk space”
  Echo “2. Display current users”
  Echo“3.Displaymemoryusage”
  Echo “4. Exit”
  Read-p“Enteryourchoice:“choice
  Case $choice in
    1) df-h;;
    2) who ;;
    3) free-h;;
    4) break;;
    *)echo“Invalidchoice.Tryagain.”;;
  Esac
Done
```

Experiments 15:

15. WriteaC-programtoforkachildprocessandexecutethegivenLinux commands.

```
#include <stdio.h>
#include <unistd.h>
#include<sys/types.h>

Intmain() {
    Pid_tpid=fork(); If
    (pid == 0) {
        // Child
        processExeclp(“/bin/ls”,“ls”,“-
        l”,NULL);
    }elseif(pid > 0){
        //Parentprocess
        Wait(NULL);
        Printf(“Childprocessexecuted.\n”);
    } else{
        Perror(“Forkfailed”);
    }
    Return 0;
}
```

Experiments 16:

16. WriteaC-programtoforkachildprocess, printownerprocessIDandits parent process ID.

```
#include <stdio.h>
#include <unistd.h>
#include<sys/types.h>

Intmain() {
    Pid_tpid=fork(); If
    (pid == 0) {
        //Childprocess
        Printf(“Child Process ID: %d\n”, getpid());
        Printf(“ParentProcessID:%d\n”,getppid());
    }elseif(pid > 0){
        //Parentprocess
        Printf(“ParentProcessID:%d\n”, getpid());
    } else{
        Perror(“Forkfailed”);
    }
    Return 0;
}
```

Experiments 17:

17. WriteaC-programtopromptttheuserforthenameoftheenvironment variable, check its validity and print an appropriate message.

```
#include <stdio.h>
#include<stdlib.h>

Intmain() {
   Char var[100];
   Printf(“Enterthenameoftheenvironmentvariable: “);
   Scanf(“%s”, var);
   Char*value=getenv(var); If
   (value) {
      Printf(“Valueof%s:%s\n”,var,value);
   } else{
      Printf(“Environmentvariable%sdoesnotexist.\n”, var);
   }
   Return 0;
}
```

Experiments 18:

18. Write a C-programtoREADdetailsofNstudentssuchasstudentname,reg number, semester and age. Find the eldest of them and display his details.

```
#include <stdio.h>
#include<string.h>

//Structuretoholdstudentdetails
Typedef struct {
    Charname[50];
    Intreg_number;
    Int semester;Int
    age;
}Student;

Intmain(){ Int
    n;

    // Input the number of students
    Printf("Enterthenumberofstudents:");
    Scanf("%d", &n);

    Studentstudents[n],eldest;
    Eldest.age=0;//Initializeeldestageto0
```

```
    // Input student details
    For(inti =0;i<n;i++){
        Printf(“\nEnterdetailsforstudent%d:\n”,i+1);
        Printf(“Name: “);
        Scanf(“%[^\n]%*c”,students[i].name);//Readstringwithspaces Printf(“Registration
        Number: “);
        Scanf(“%d”,&students[i].reg_number);
        Printf(“Semester: “);
        Scanf(“%d”,&students[i].semester);
        Printf(“Age: “);
        Scanf(“%d”,&students[i].age);

        //Checkifthecurrentstudentisolder If
        (students[i].age > eldest.age) {
            Eldest= students[i];
        }
    }

    //Displaythedetailsoftheeldeststudent
    Printf(“\nThe eldest student is:\n”);
    Printf(“Name: %s\n”, eldest.name);
    Printf(“RegistrationNumber:%d\n”,eldest.reg_number);
    Printf(“Semester: %d\n”, eldest.semester);
    Printf(“Age:%d\n”,eldest.age);
    Return 0;
}
```

THE END