



LAB ASSIGNMENT

Operating Systems Lab CSE 324

Submitted To:

Most. Hasna Hena
Assistant Professor
Department of CSE
Daffodil International University

Submitted By:

Fahim Ur Rahman
ID: 201-15-13674

Section: B

Submitted On: 7th December, 2022

Problem 1 Solution:

```
1  #!/bin/bash
2
3  echo "Showing Current Directory:"
4  pwd
5
6  echo "Showing List Of Files in the Directory:"
7  ls
8
9  echo "Changing Directory to Desktop:"
10 cd Desktop
11 mkdir folder201-15-13674
12 touch file1.txt
13 touch file2.txt
14 cat file1.txt
15 cat file2.txt
16
17 echo "Creating Another file to Write something about myself:"
18 touch aboutme
```

Problem 2 Solution:

```
1  #!/bin/bash
2
3  echo "Welcome "
4  u=$USER
5  echo "User: $u"
6
```

Problem 3 Solution:

```
1  #!/bin/bash
2
3  num1=10
4  num2=20
5
6  sum=$((num1 + num2))
7  echo -e "Summation:"
8  echo $sum
9
10 sub=$((num1 - num2))
11 echo -e "Subtraction:"
12 echo $sub
13
14 mul=$((num1 * num2))
15 echo -e "Multiplication:"
16 echo $mul
17
18 div=$((num1 / num2))
19 echo -e "Division:"
20 echo $div
```

Problem 4 Solution:

```
. > Semesters > 5th Semester > Operating
1  #!/bin/bash
2
3  echo "--Even Or Odd--"
4  echo -n "Enter Number:"
5  read n
6  echo -n "Result:"
7  if [ `expr $n % 2` == 0 ]
8  ✓ then
9      |   echo "$n is Even"
10 ✓ else
11     |   echo "$n is Odd"
12 fi
13
```

Problem 5 Solution:

```
1  #!/bin/bash
2
3  read -p "Enter Grade: " grade
4
5  if [ $grade -le 50 ]
6  ✓ then
7      |   echo "Grade:\t Fail"
8  elif [ $grade -le 59 ]
9  ✓ then
10     |   echo "Grade:\t B"
11  elif [ $grade -le 79 ]
12  ✓ then
13     |   echo "Grade:\t A"
14  elif [ $grade -le 100 ]
15  ✓ then
16     |   echo "Grade:\t A+"
17  elif [ $grade -gt 100 ]
18  ✓ then
19  ✓ |   echo "Invalid!!"
20     |   exit
21 fi
```

Problem 6 Solution:

```
1  #!/bin/bash
2
3  echo "Enter Number: "
4  read num
5
6  fact=1
7
8  while [ $num -gt 1 ]
9  do
10     fact=$((fact * num))
11     num=$((num - 1))
12 done
13
14 echo -e "Factorial:"
15 echo $fact
16 |
```

Problem 7 Solution:

```
1  #!/bin/bash
2
3  echo "Enter Value of N:"
4  read n
5  a=0
6  b=1
7  count=2
8  echo "Fibonacci Series:"
9  echo $a
10 echo $b
11 while [ $count -le $n ]
12 do
13     fib=`expr $a + $b`
14     a=$b
15     b=$fib
16     count=`expr $count + 1`
17 done
```

Problem 8 Solution:

```
1  #!/bin/bash
2
3  echo "Enter Number:"
4  read number
5  rows=$number
6  for((i=1; i<=rows; i++))
7  do
8      for((j=1; j<=i; j++))
9      do
10         echo -n "$i "
11         number=$number
12     done
13     echo
14     number=$((number + 1))
15 done
```

Problem 9 Solution:

```
1  #!/bin/bash
2
3  series(){
4      echo "Enter Number:"
5      read num
6      num2=22
7      for (( i=$num2; i<=num; i+=20 ))
8      do
9          sum=$(( $num2 + $i ))
10         echo "Sum: $i"
11         sum2=$(( $sum2 + $sum ))
12     done
13     echo "Summation of Series: $sum2 "
14 }
15
16 echo "Calling Function:"
17 series
```

Problem 10 Solution:

```
1  #!/bin/bash
2
3  palindrome(){
4  echo "Enter Number to Check:"
5  read num
6  s=0
7  rev=""
8  temp=$num
9
10 while [ $num -gt 0 ]
11 do
12     s=$(( $num % 10 ))
13     num=$(( $num / 10 ))
14     rev=$( echo ${rev}${s} )
15 done
16
17 if [ $temp -eq $rev ];
18 then
19     echo "Number is Palindrome"
20 else
21     echo "Number is Not Palindrome!!!"
22 fi
23 }
24
25
26 while [ 1 ]
27 do
28     echo "Enter Choice : y to continue, n to stop"
29     read choice
30     if [ $choice == 'n' ]
31     then
32         exit
33     else
34         palindrome
35     fi
36 done
37
```

Problem 11 Solution:

```
#include<stdio.h>
int main()
{
    int
process[10],arrival[10],burst[10],start[10],end[10],wait[10],turnaround[10];
    int n,i,j,temp,totalwait=0,totalturn=0;
    float avgwait,avgturn;
    printf("Enter the number of processes\n");
    scanf("%d",&n);
    printf("Enter value for each process\n\n");
    printf("process name,arrival time,Burst time\n");
    for(i=0;i<n;i++)
        scanf("%d%d%d",&process[i],&arrival[i],&burst[i]);
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(arrival[i]>arrival[j])
            {
                temp=arrival[i];
                arrival[i]=arrival[j];
                arrival[j]=temp;
                temp=process[i];
                process[i]=process[j];
                process[j]=temp;
                temp=burst[i];
                burst[i]=burst[j];
                burst[j]=temp;
            }
        }
    }
    printf("\nProcess name\tArrival time\tBurst time\n");
    for(i=0;i<n;i++)
        printf("p[%d]\t\t%d\t\t%d\n",process[i],arrival[i],burst[i]);
    start[0]=arrival[0];
    end[0]=arrival[0]+burst[0];
    wait[0]=0;
    turnaround[0]=burst[0];
    totalturn+=turnaround[0];
    for(i=1;i<n;i++)
    {
        if(arrival[i]<=end[i-1])
            start[i]=end[i-1];
        else
            start[i]=arrival[i];
        wait[i]=start[i]-arrival[i];
```

```

        totalwait+=wait[i];
        end[i]=start[i]+burst[i];
        turnaround[i]=burst[i]+wait[i];
        totalturn+=turnaround[i];
    }
    avgwait=(float)totalwait/(float)n;
    avgturn=(float)totalturn/(float)n;
    printf("\n\n|Process name\t|Arrival time\t|Burst time\t|Start
time\t|End time\t|Waiting time\t|Turnaround time|\n");
    for(i=0;i<n;i++)
        printf("|p[%d]\t\t|d\t\t|d\t\t|d\t\t|d\t\t|d\t\t|d\n",process[
i],arrival[i],burst[i],start[i],
                end[i],wait[i],turnaround[i]);
    printf("Total waiting time is %d\n",totalwait);
    printf("Average waiting time is %f\n",avgwait);
    printf("Total turnaround time is %d\n",totalturn);
    printf("Average turnaround time is %f\n",avgturn);
    return 0;
}

```


Problem 12 Solution:

```
#include<stdio.h>
int main()
{
    int at[10],bt[10],pr[10];
    int
n,i,j,temp,time=0,count,over=0,sum_wait=0,sum_turnaround=0,start;
    float avgwait,avgturn;
    printf("Enter the number of processes\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter the arrival time and burst time for
process %d\n",i+1);

        scanf("%d%d",&at[i],&bt[i]);
        pr[i]=i+1;
    }
    for(i=0;i<n-1;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if(at[i]>at[j])
            {
                temp=at[i];
                at[i]=at[j];
                at[j]=temp;
                temp=bt[i];
                bt[i]=bt[j];
                bt[j]=temp;
                temp=pr[i];
                pr[i]=pr[j];
                pr[j]=temp;
            }
        }
    }
    printf("\n\nProcess\t|Arrival time\t|Burst time\t|Start time\t|End
time\t|waiting time\t|Turnaround time\n\n");
    while(over<n)
    {
        count=0;
        for(i=over;i<n;i++)
        {
            if(at[i]<=time)
            count++;
            else
            break;
        }
    }
```

```

        if(count>1)
        {
            for(i=over;i<over+count-1;i++)
            {
                for(j=i+1;j<over+count;j++)
                {
                    if(bt[i]>bt[j])
                    {
                        temp=a
t[i];
                        at[i]=
at[j];
                        at[j]=
temp;
                        temp=b
t[i];
                        bt[i]=
bt[j];
                        bt[j]=
temp;
                        temp=p
r[i];
                        pr[i]=
pr[j];
                        pr[j]=
temp;

                    }
                }
            }
            start=time;
            time+=bt[over];
            printf("p[%d]\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t|\t
%d\n",pr[over],
                        at[over],bt[over],start,time,time-
at[over]-bt[over],time-at[over]);
            sum_wait+=time-at[over]-bt[over];

            sum_turnaround+=time-at[over];
            over++;
        }
        avgwait=(float)sum_wait/(float)n;
        avgturn=(float)sum_turnaround/(float)n;
        printf("Average waiting time is %f\n",avgwait);
        printf("Average turnaround time is %f\n",avgturn);
        return 0;
    }

```