8/4/2018

**Lab Report**

Subject: Operating System Lab

Submitted To:

Md. Ohidujjaman Tuhin

Lecturer at DIU

Submitted By:

Md. Ahosan Habib

ID: 161-15-974. Section: PC-A.

1. **Linear Search**

clear

echo "How many value you have:"

read n

for((i=0;i<n;i++))

do

read a[$i]

done

echo "Enter search item:"

read item

for((i=0;i<n;i++))

do

if [ $item -eq ${a[$i]} ]

then

echo "item found in index number : $i"

fi

done

1. **Binary:**

echo "Enter the number of items"

read n

echo "Enter items in ascending order"

for((i=0;i<$n;i++))

do

read A[$i]

done

echo "Enter search item"

read i

tem

first=0

last=`expr $n - 1`

while (($first<=$last))

do

x=`expr $first + $last`

mid=`expr $x / 2`

if(($item==${A[$mid]}))

then

flag=1

break

else

if(($item>${A[$mid]}))

then

first=`expr $mid + 1`

else

last=`expr $mid - 1`

fi

fi

done

if [[ $flag = 1 ]]

then

echo "Itemm found"

else

echo "Item not found"

fi

1. **Prime number:**

#! /bin/bash

read n

for ((i=2;i<n;i++))

do

if [ $i == 2 ]

then

echo "2"

fi

for((j=2;j<i;j++))

do

a=`expr $i % $j`

b=`expr $j + 1`

if [ $a == 0 ]

then

break

elif [ $i == $b ]

then

echo "$i"

fi

done

done

1. **Fibonacci:**

clear

echo "Enter how many index you want to check:"

read val

a=1

b=-1

for((i=0;i<val;i++))

do

c=`expr $a + $b`

echo $c

b=$a

a=$c

done

**5. Celsius to ferenhit and ferenhit to celsius:**

echo " enter 1 for c to f || enter 2 for f to c"

read i

if(($i==1))

then

echo "enter c"

read c

x=`expr $c / 5`

y=`expr $x \\* 9`

f=`expr $y + 32`

echo " temprature in f= $f"

else

echo "read f"

read f

x=`expr $f - 32`

y=`expr $s / 9`

c=`expr $y \\* 5`

echo " value in c= $c"

fi

**6. Arithmetic:**

echo "enter value a "

read a

echo "enter value b "

read b

c=`expr $a + $b`

echo "sum is $c"

c=`expr $a - $b`

echo "sub is $c"

c=`expr $a \\* $b`

echo "mul is $c"

c=`expr $a / $b`

echo "div is $c"

**7. Smallest number:**

echo "enter a"

read a

echo "enter b"

read b

echo "enter c"

read c

if [ $a -lt $b ]

then

if [ $a -lt $c ]

then

echo " $a is smallest"

exit

fi

echo " $c is smallest"

fi

if [ $b -lt $a ]

then

if [ $b -lt $c ]

then

echo " $b is smallest"

fi

echo " $c is smallest"

fi

**8. smallest number 2:**

echo "enter a"

read a

echo "enter b"

read b

echo "enter c"

read c

if [ $a -lt $b ]

then

if [ $a -lt $c ]

then

echo " $a is smallest"

exit

fi

echo " $c is smallest"

else

if [ $b -lt $a ]

then

if [ $b -lt $c ]

then

echo " $b is smallest"

else

echo " $c is smallest"

fi

fi

fi

**9. FCFS:**

echo "Enter process number n:"

read n

echo "Enter AT number:"

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

do

read AT[$i]

done

echo "Enter BT number:"

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

do

read BT[$i]

done

TWT=0

CT=${AT[1]}

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

do

CT=`expr $CT + ${BT[$i]}`

TAT=`expr $CT - ${AT[$i]}`

WT=`expr $TAT - ${BT[$i]}`

TWT=`expr $TWT + $WT`

done

AWT=`expr $TWT / $n`

echo "The average WT is: $AWT"

**10. SJF:**

#! /bin/bash

echo "How many process you want to give: "

read n

echo "Enter $n Arrival Time: "

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

do

read at[i]

done

echo "Enter $n Burst Time: "

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

do

read bt[i]

done

#bubble sort for arrival time

for((c=1;c<=$n;c++))

do

for((d=1;d<=$n-$c;d++))

do

if [ ${at[$d]} > ${at[$d+1]} ]

then

swap=${at[$d]}

at[d]=${at[$d+1]}

at[d+1]=$swap

fi

done

done

ct[1]=${at[1]}

twt=0

j=1

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

do

ct[i]=`expr ${ct[$j]} + ${bt[$i]}`

tat[i]=`expr ${ct[$i]} - ${at[$i]}`

wt[i]=`expr ${tat[$i]} - ${bt[$i]}`

twt=`expr $twt + ${wt[$i]}`

j=$i

#bubble Sort

for((c=0;c<=$n-1;c++))

do

for((d=2;d<=$n-$c-1;d++))

do

if [ ${bt[$d]} > ${bt[$d+1]} ]

then

swap=${bt[$d]}

bt[d]=${bt[$d+1]}

bt[d+1]=$swap

fi

done

done

done

echo "AT: "

for((i=1;i<=n;i++))

do

echo "${at[$i]}"

done

echo "BT: "

for((i=1;i<=n;i++))

do

echo "${bt[$i]}"

done

echo "CT: "

for((i=1;i<=n;i++))

do

echo "${ct[$i]}"

done

echo "TAT: "

for((i=1;i<=n;i++))

do

echo "${tat[$i]}"

done

echo "WT: "

for((i=1;i<=n;i++))

do

echo "${wt[$i]}"

done

echo "Average waiting time:"

#avgwt=`expr $twt / $n`

echo "scale=3;$twt/$n" | bc

**11.priority:**

#! /bin/bash

echo "How many process you want to give: "

read n

echo "Enter $n Arrival Time: "

for ((i=0;i<$n;i++))

do

read at[i]

done

echo "Enter $n Burst Time: "

for ((i=0;i<$n;i++))

do

read bt[i]

done

echo "Enter $n priority "

for ((i=0;i<$n;i++))

do

read pr[i]

done

# Bubble Sort for AT

for((i=0;i<$n-1;i++))

do

for((j=0;j<$n-$i-1;j++))

do

if [ ${at[$j]} -gt ${at[$j+1]} ]

then

swap=${at[$j]}

at[j]=${at[$j+1]}

at[j+1]=$swap

swap1=${bt[$j]}

bt[j]=${bt[$j+1]}

bt[j+1]=$swap1

swap2=${pr[$j]}

pr[j]=${pr[$j+1]}

pr[j+1]=$swap2

fi

done

done

#SJF

ct[0]=${at[0]}

twt=0

j=0

for ((i=0;i<$n;i++))

do

ct[i]=`expr ${ct[$j]} + ${bt[$i]}`

tat[i]=`expr ${ct[$i]} - ${at[$i]}`

wt[i]=`expr ${tat[$i]} - ${bt[$i]}`

twt=`expr $twt + ${wt[$i]}`

j=$i

#bubble Sort for pr

for((c=1;c<$n;c++))

do

for((d=1;d<$n-$c;d++))

do

if [ ${pr[$d]} -gt ${pr[$d+1]} ]

then

swap3=${pr[$d]}

pr[d]=${pr[$d+1]}

pr[d+1]=$swap3

swap=${bt[$d]}

bt[d]=${bt[$d+1]}

bt[d+1]=$swap

swap=${at[$d]}

at[d]=${at[$d+1]}

at[d+1]=$swap

swap3=${pr[$d]}

pr[d]=${pr[$d+1]}

pr[d+1]=$swap3

fi

done

done

done

#avgwt=`expr $twt / $n`

echo "scale=3;$twt/$n" | bc

# here bc is used for float number

**12.for:**

read n

sum=0

mul=1

for (( i=1; i<=n; i++))

do

mul=`expr $i \\* $i`

sum=`expr $sum + $mul`

echo "$sum"

done

**13.physical address:**

echo "enter size"

read n

echo "enter $n segment"

for((i=0;i<$n;i++))

do

read s[$i]

done

echo "enter $n limit"

for((i=0;i<$n;i++))

do

read l[$i]

done

echo "enter $n burst time"

for((i=0;i<$n;i++))

do

read b[$i]

done

echo "enter segment to findd physical address"

read seg

echo "read logical address for segment $seg"

read lo

for((i=0;i<$n;i++))

do

if((${s[$i]}==$seg))

then

if(($lo<=${l[$i]}))

then

x=`expr ${b[$i]} + $lo`

echo "pysical address = $x"

exit

fi

fi

done

if [ i>$n ]

then

echo "Interrupt"

fi

**14.Bankers:**

#!/bin/bash

declare -A allocation

declare -A claim

declare -A need

declare -A ia

declare -A ta

declare -A ca

echo "read number of row"

read m

echo "read column"

read n

echo "read allocation matrix"

for ((i=1;i<=m;i++)) do

for ((j=1;j<=n;j++)) do

read x

allocation[$i,$j]=$x

done

done

echo "read claime matrix"

for ((i=1;i<=m;i++)) do

for ((j=1;j<=n;j++)) do

read y

claim[$i,$j]=$y

done

done

for ((i=1;i<=m;i++)) do

for ((j=1;j<=n;j++)) do

p=${claim[$i,$j]}

q=${allocation[$i,$j]}

r=`expr $p - $q`

need[$i,$j]=$r

done

done

echo "read initial availibility"

for ((i=1;i<=1;i++)) do

for ((j=1;j<=n;j++)) do

read x

ia[$i,$j]=$x

done

done

for((i=1;i<=n;i++))

do

al[i]=0

done

for ((i=1;i<=n;i++)) do

for ((j=1;j<=m;j++)) do

x=${allocation[$j,$i]}

al[i]=`expr ${al[i]} + $x`

done

done

#read total allocation in a matrix

echo "total allocation "

for ((i=1;i<=1;i++)) do

for ((j=1;j<=n;j++)) do

ta[$i,$j]=${al[$j]}

done

done

for ((i=1;i<=1;i++)) do

for ((j=1;j<=n;j++)) do

p=${ia[$i,$j]}

q=${ta[$i,$j]}

r=`expr $p - $q`

ca[$i,$j]=$r

done

done

for ((i=1;i<=m;i++)) do

for ((j=1;j<=n;j++)) do

echo "${need[$i,$j]}"

done

done

echo "total allocation"

for ((i=1;i<=1;i++)) do

for ((j=1;j<=3;j++)) do

echo "${ta[$i,$j]}"

done

done

echo "current availibility"

for ((i=1;i<=1;i++)) do

for ((j=1;j<=3;j++)) do

echo "${ca[$i,$j]}"

done

done

**15. SSTF:**

#! /bin/bash

echo "Enter Total Cylionders: "

read total

echo "Enter Header: "

read header

echo "Enter Previous Request: "

read req

echo "Enter How Many Values You Want To Give: "

read n

echo "Enter $n values: "

for ((i=0;i<n;i++))

do

read q[i]

#temp[i]=${q[$i]}

done

n=`expr $n + 1`

q[n-1]=$header

# Bubble Sort for Q

for((i=0;i<$n-1;i++))

do

for((j=0;j<$n-$i-1;j++))

do

if [ ${q[$j]} -gt ${q[$j+1]} ]

then

swap=${q[$j]}

q[j]=${q[$j+1]}

q[j+1]=$swap

fi

done

done

echo "After Sorting: "

for ((i=0;i<$n;i++))

do

echo "${q[$i]}"

done

seek=0

for ((i=0;i<n;i++))

do

if [ $header == ${q[$i]} ]

then

aa=`expr $header - ${q[$i-1]}`

bb=`expr $header - ${q[$i+1]}`

if [ $aa -le 0 ]

then

m=-1

aa=`expr $aa \\* $m`

elif [ $bb -le 0 ]

then

m=-1

bb=`expr $bb \\* $m`

fi

k=$i

if [ $aa -le $bb ]

then

for ((j=0;j<n;j++))

do

cal=`expr $header - ${q[$k-1]}`

seek=`expr $seek + $cal`

k=`expr $k - 1`

header=${q[$k]}

if [ $header == ${q[1]} ]

then

cal=`expr $header - ${q[0]}`

seek=`expr $seek + $cal`

header=${q[0]}

break

fi

done

k=$i

for ((j=0;j<n;j++))

do

cal=`expr ${q[$k+1]} - $header`

seek=`expr $seek + $cal`

k=`expr $k + 1`

header=${q[$k]}

if [ $header == ${q[$n-2]} ]

then

cal=`expr ${q[$n-1]} - $header`

seek=`expr $seek + $cal`

#header=${q[$n-1]}

break

fi

done

fi

fi

done

echo "Total Seek Time: $seek"

**16.Factorial:**

echo "enter a number"

read a

fact=1

while [ $a -gt 0 ]

do

fact=`expr $a \\* $fact`

a=`expr $a - 1`

done

echo "Factorial ia: $fact"

**17. Round roben:**

#! /bin/bash

echo "Enter How Many Process You Want To Give: "

read n

echo "Enter Time Quantam: "

read tq

echo "Enter $n AT: "

for ((i=0;i<n;i++))

do

read at[i]

done

echo "Enter $n BT: "

for ((i=0;i<n;i++))

do

read bt[i]

done

# bubble sort for AT

for ((i=0;i<$n-1;i++))

do

for ((j=0;j<$n-$i-1;j++))

do

if [ ${at[$j]} -gt ${at[$j+1]} ]

then

swap=${at[$j]}

at[j]=${at[$j+1]}

at[j+1]=$swap

swap=${bt[$j]}

bt[j]=${bt[$j+1]}

bt[j+1]=$swap

fi

done

done

# calculation

ct[0]=${at[0]}

twt=0

j=0

c=0

for ((i=0;i<n;i++))

do

if [ ${bt[$i]} -gt $tq ]

then

cw[i]=$tq

n=`expr $n + 1`

bt[n-1]=`expr ${bt[$i]} - $tq`

else

cw[i]=${bt[$i]}

fi

j=$i

done

echo "Context Switching Values : "

for ((i=0;i<n;i++))

do

echo "${cw[$i]}"

done

18.