|  |  |  |  |
| --- | --- | --- | --- |
|  | **COLLEGE OF COMPUTING AND INFORMATION SCIENCES** | | |
| **Assignment # 01** | | |
| **Course Title** | Operating System | **Total Marks** | 10 |
| **Date** | March 07, 2022 | **Class ID** | 109134 |
| **Student Id** | **11067** | **Student Name** | **SYED MUZZAMIL WASEEM** |

# Instructions:

* Copied work and late submission will be marked as ZERO.
* Attach your code and screenshot of your output in this file.
* Submit hardcopy of your solution in class.

**Submission Deadline: 07-03-2022**

# Question 1:

Write a program that allocate memory for array and print the elements of array (size and elements of array should be user defined), search the number form array using any searching algorithm.

Next, reallocate the size of array and print the array elements. For memory reallocation write down your own **realloc** function using **malloc** and **free** function. Suppose if your array size is of 5 indexes then the minimum memory required is of 20bytes. After reallocation you can increase or decrease the memory. If the memory size is incremented print 0 at new indexes or if the memory size is decremented, then remove the indexes from array.

**[Add code and screenshot of output here]**

**Code:**

# #include <stdio.h>

# #include <stdlib.h>

# int main(){

# int s,\*ptr,total,t,f,item;

# printf("Size of the Array: ");

# scanf("%d",&s);

# ptr=(int\*)malloc(s\*sizeof(int));

# for(int i=0;i<s;i++){

# printf("Element for %02d: ",i+1);

# scanf("%d",(ptr+i));}

# printf("\n Search number from array: ");

# scanf("%d",&item);

# for(int i=0; i<s; i++){

# if(\*(ptr+i)==item){

# f=1;

# break;}}

# if(f==1){

# printf("Number is available");}

# else{

# printf("Number is not available!");}

# printf("\nArray contains :\n");

# for (int i = 0; i < s; i++) {

# for (int j = i + 1; j < s; j++) {

# if (\*(ptr + j) < \*(ptr + i)) {

# t = \*(ptr + i);

# \*(ptr + i) = \*(ptr + j);

# \*(ptr + j) = t;

# }}}

# for(int i=0;i<s;i++){

# printf("%d\n",\*(ptr+i));}

# printf("Reallocating memory size: ");

# scanf("%d",&s);ptr = (int\*)realloc(ptr, s \* sizeof(int));

# printf("After reallocation array contains : \n");

# for(int i=0;i<s;i++){

# printf("%d\n",\*(ptr+i));}

# printf("Reallocating memory size again: ");

# scanf("%d",&s);

# ptr = (int\*)realloc(ptr, s \* sizeof(int));

# //read array elements

# for(int i=0;i<s;i++){

# printf("Element for %02d: ",i+1);

# scanf("%d",(ptr+i));}

# printf("\nArray contains :\n");

# for (int i = 0; i < s; i++) {

# for (int j = i + 1; j < s; j++) {

# if (\*(ptr + j) < \*(ptr + i)) {

# t= \*(ptr + i);

# \*(ptr + i) = \*(ptr + j);

# \*(ptr + j) = t;}}}

# for(int i=0;i<s;i++){

# printf("%d\n",\*(ptr+i));}

# printf("Reallocating memory size: ");

# scanf("%d",&s);

# ptr = (int\*)realloc(ptr, s \* sizeof(int));

# printf("After reallocation array contains : \n");

# for(int i=0;i<s;i++){

# printf("%d\n",\*(ptr+i));}

# return 0;}

# Screenshots:

A screenshot of a computer

Description automatically generated

**Question 2:**

Write a program that implements **Process Scheduling Algorithm** (FCFS, SJF – Preemptive , SJF

– Non Preemptive). First your program need to ask what scheduling algorithm you need to execute. After selecting process scheduling algorithm, ask user to input Process ID, Arrival Time and Burst Time. In the end, Calculate Start Time, Completion Time, Waiting Time, Turnaround Time, Average Waiting Time and Average Turnaround Time.

Print all the values in the tabular form.

**[Add code and screenshot of output here]**

**Code:**

# #include <stdio.h>

# #include <string.h>

# int main() {

# int alg;

# printf("\n<--- SYED MUZZAMIL WASEEM - 11067 --->");

# printf("\nSelect Algorithm Among: \n\t1.FCFS \n\t2.SJF-Preemptive \n\t3.SJF-Non Preemptive \nInsert Number: ");

# scanf("%d",&alg);

# if (alg==1){

# char pn[10][10],t[10];

# int arr[10],bt[10],st[10],ct[10],tat[10],wt[10],i,j,n,temp,totwt=0,tottat=0;

# printf("\nPlease Enter the no of processes: ");

# scanf("%d",&n);

# printf("\n");

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

# printf("Enter the ProcessID, Arrival Time & Burst Time : ");

# scanf("%s%d%d",&\*pn[i],&arr[i],&bt[i]);}

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

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

# if(arr[i]<arr[j]){

# temp=arr[i]; arr[i]=arr[j]; arr[j]=temp; temp=bt[i]; bt[i]=bt[j]; bt[j]=temp;

# strcpy(t,pn[i]); strcpy(pn[i],pn[j]); strcpy(pn[j],t);}}}

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

# if(i==0) st[i]=arr[i];

# else

# st[i]=ct[i-1];

# wt[i]=st[i]-arr[i];

# ct[i]=st[i]+bur[i];

# tat[i]=ct[i]-arr[i];}

# printf("\nPID\tAT\tBT\tST\tCT\tTAT\tWT");

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

# printf("\n%s\t%3d\t%3d\t%3d\t%6d\t%6d\t%3d",pn[i],arr[i],bt[i],st[i],ct[i],tat[i],wt[i]);

# totwt+=wt[i]; tottat+=tat[i];}

# printf("\nAverage Waiting Time is:\t%f",(float)totwt/n);

# printf("\nAverage Turn Around Time is:\t%f\n",(float)tottat/n);}

# else if (algo==2){

# int bt[20],at[10],n,i,j,temp,st[10],ct[10],wt[10],tat[10],totwt=0,totta=0;

# float awt,ata;

# char pn[10][10],t[10];

# printf("\nPlease Enter the no of processes: ");

# scanf("%d",&n);

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

# printf("Enter the ProcessID, Around Time & Burst Time : ");

# scanf("%s%d%d",pn[i],&at[i],&bt[i]);}

# for(i=0; i<n; i++) for(j=0; j<n; j++){

# if(bt[i]<bt[j]){

# temp=at[i]; at[i]=at[j]; at[j]=temp; temp=bt[i]; bt[i]=bt[j]; bt[j]=temp;

# strcpy(t,pn[i]); strcpy(pn[i],pn[j]); strcpy(pn[j],t);}}

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

# if(i==0)

# st[i]=at[i];

# else

# st[i]=ct[i-1];

# wt[i]=st[i]-at[i];

# ct[i]=st[i]+bt[i];

# tat[i]=ct[i]-at[i]; totwt+=wt[i]; totta+=tat[i];}

# awt=(float)totwt/n; ata=(float)totta/n;

# printf("\nProcessID\Turn Around Time\Total Burst Time\Total Start Time\Total Completion Time\Total Turn Around Time\Total Waiting Time");

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

# printf("\n%s\t%5d\t%5d\t%5d\t%5d\t%5d\t%5d",pn[i],at[i],bt[i],st[i],ct[i],tat[i],wt[i]);

# printf("\nAverage WT is:\t%f",awt);

# printf("\nAverage Turn Around Time is:\t%f",ata);}

# else if(algo==3){

# int i,n,p[10]={1,2,3,4,5,6,7,8,9,10},min,k=1,btime=0;

# int bt[10],temp,j,at[10],wt[10],tt[10],ta=0,sum=0;

# float wavg=0,tavg=0,tsum=0,wsum=0;

# printf("\nPlease Enter the no of processes: ");

# scanf("%d",&n);

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

# printf("Enter AT & BT : ");

# scanf("%d%d",&at[i],&bt[i]);}

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

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

# if(at[i]<at[j]){

# temp=p[j]; p[j]=p[i]; p[i]=temp; temp=at[j]; at[j]=at[i]; at[i]=temp; temp=bt[j]; bt[j]=bt[i]; bt[i]=temp;}}}

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

# btime=btime+bt[j]; min=bt[k]; for(i=k;i<n;i++){

# if (btime>=at[i] &&bt[i]<min){

# temp=p[k]; p[k]=p[i]; p[i]=temp; temp=at[k]; at[k]=at[i]; at[i]=temp; temp=bt[k]; bt[k]=bt[i]; bt[i]=temp;}}

# k++;}

# wt[0]=0;

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

# sum=sum+bt[i-1]; wt[i]=sum-at[i]; wsum=wsum+wt[i];}

# wavg=(wsum/n);

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

# ta=ta+bt[i]; tt[i]=ta-at[i]; tsum=tsum+tt[i];}

# tavg=(tsum/n);

# printf("\nProcessID\Total Awaiting Time\Total Burst Time\Total Turn Around Time\Total Waiting Time");

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

# printf("\n%d\t%d\t%d\t%d\t%d",p[i],at[i],bt[i],tt[i],wt[i]);}

# printf("\n\nAverage Waiting Time is:\t%f",wavg);

# printf("\nAverage Turn Around Time is:\t%f\n",tavg);}

# else{

# printf("No other options avialable");}

# return 0;}

# Screenshots:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated