WEEK 8

Yash Gupta
1BM21CS251

02-08-2023

Q: Write a C program to simulate the following contiguous memory allocation techniques
a) Worst-fit
b) Best-fit
c) First-fit

Example

Consider a swapping system in which memory consists of the following whole sizes in memory order: 10K, 4k, 20k, 18k, 7k, 9k, 12k, and 15k. Which hole is taken for successive segment request of i)12k, ii)10k, iii)9k for first fit? Now repeat the question for best fit and worst fit.

First Fit		
12k	→	20k
10k	→	10k
9k	→	18k

Best Fit		
12k	→	12k
10k	→	10k
9k	→	9k

Worst Fit		
12k	→	20k
10k	→	18k
9k	→	15k

```
#include <stdio.h>
int holes[10];
int holes_free[10];
int request[10];
int completed[10];
int h,n,temp;
void best_fit(int n, int h){
    for(int i=0;i<n;i++){
        completed[i]=0;
    }
    for(int i=0;i<h;i++){
        holes_free[i]=1;
    }
    printf("Best Fit:\n");
    for(int i=0;i<h-1;i++){
        for(int j=0;j<h-i-1;j++){
            if(holes[j+1]<holes[j]){
                temp=holes[j+1];
                holes[j+1]=holes[j];
                holes[j]=temp;
            }
        }
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<h;j++){
            if(request[i]<=holes[j] && holes_free[j]==1 && completed[i]==0){
                completed[i]=1;
                holes_free[j]=0;
                printf("%dk in %dk\n",request[i],holes[j]);
            }
        }
    }
}
```

```
void worst_fit(int n, int h){
   for(int i=0;i<n;i++){
      completed[i]=0;
   }
   for(int i=0;i<h;i++){
      holes_free[i]=1;
   }
   printf("Worst Fit:\n");
   for(int i=0;i<h-1;i++){
      for(int j=0;j<h-i-1;j++){
         if(holes[j+1]>holes[j]){
            temp=holes[j+1];
            holes[j+1]=holes[j];
            holes[j]=temp;
         }
      }
   }
   for(int i=0;i<n;i++){
      for(int j=0;j<h;j++){
         if(request[i]<=holes[j] && holes_free[j]==1 && completed[i]==0){
            completed[i]=1;
            holes_free[j]=0;
            printf("%dk in %dk\n",request[i],holes[j]);
         }
      }
   }
}
void first_fit(int n, int h){
   for(int i=0;i<n;i++){
      completed[i]=0;
   }
   for(int i=0;i<h;i++){
      holes_free[i]=1;
```

```
    }
    printf("First Fit:\n");
    for(int i=0;i<n;i++){
        for(int j=0;j<h;j++){
            if(request[i]<=holes[j] && holes_free[j]==1 && completed[i]==0){
                completed[i]=1;
                holes_free[j]=0;
                printf("%dk in %dk\n",request[i],holes[j]);
            }
        }
    }
}
int main(){
    printf("enter the number of holes:\t");
    scanf("%d",&h);
    printf("Enter the holes sizes:\n");
    for(int i=0;i<h;i++){
        scanf("%d",&holes[i]);
    }
    printf("enter the number of requests:\t");
    scanf("%d",&n);
    printf("Enter the request segments:\n");
    for(int i=0;i<n;i++){
        scanf("%d",&request[i]);
    }
    best_fit(n,h);
    worst_fit(n,h);
    first_fit(n,h);
    return 0;
}
```

OUTPUT:

```
enter the number of holes:      8
Enter the holes sizes:
10
4
20
18
7
9
12
15
enter the number of requests:   3
Enter the request segments:
12
10
9
Best Fit:
12k in 12k
10k in 10k
9k in 9k

Worst Fit:
12k in 20k
10k in 18k
9k in 15k

First Fit:
12k in 20k
10k in 10k
9k in 18k
```