VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

OPERATING SYSTEMS LAB 18.05.2021

MEMORY MANAGEMENT

-V.MAHEYSH

19BCE1120

Aim:

To apply various memory management algorithms.

INPUT CONSIDERED:

Memory Partition

Partition	Memory Size Available
Partition 1	100 KB
Partition 2	500 KB
Partition 3	200 KB
Partition 4	300 KB
Partition 5	600 KB

Process Requires

Process	Memory Required
Process 1	212 KB
Process 2	417 KB
Process 3	112 KB
Process 4	426 KB

FIRST FIT ALGORITHM:

```c
#include<stdio.h>


void FirstFit(int bs[50],int m,int ps[50],int n)
{
        int ab[50];
        for(int i=0;i<n;i++)
        {

                ab[i]=-1;
                for(int j=0;j<m;j++)
                {

                        if(ps[i]<=bs[j])
                        {
                                bs[j]-=ps[i];
                                ab[i]=j;
                                break;
                        }
                }
        }
        printf("\nMemory block assigned:-\n ");
        printf("   Process\tSize\tBlock\n");
        for(int i=0;i<n;i++)
        {

                printf("\t%d\t%d\t",i+1,ps[i]);
                if(ab[i]!=-1)
                        printf("%d\n",ab[i]+1);
                else
                        printf("%s\n","Not allocated");
        }

}

int main()
{
        int m,bs[50],n,ps[50];
        printf("FIRST FIT ALGORITHM:\n");
        printf("\nEnter no of blocks: ");
        scanf("%d",&m);
        printf("Enter each block size: ");
        for(int i=0;i<m;i++)
        scanf("%d",&bs[i]);
        printf("Enter no of processes: ");
        scanf("%d",&n);
        printf("Enter each process size: ");
        for(int i=0;i<n;i++)
        scanf("%d",&ps[i]);
        FirstFit(bs,m,ps,n);
}

```

OUTPUT:

```
root@kali:~/maheysh/18_05_21LAB# gcc FIRSTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
FIRST FIT ALGORITHM:

Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
Enter each process size: 212 417 112 426

Memory block assigned:-
    Process     Size    Block
        1       212     2
        2       417     5
        3       112     2
        4       426     Not allocated
```

NEXT FIT ALGORITHM:

```c
#include<stdio.h>


void NextFit(int bs[50],int m,int ps[50],int n)
{
        int ab[50],c,j;
        for(int i=0;i<n;i++)
        {

                ab[i]=-1;
                c=0;
                j=0;
                while(c<m)
                {

                        if(ps[i]<=bs[j])
                        {
                                bs[j]-=ps[i];
                                ab[i]=j;
                                break;
                        }
                        j=(j+1)%m;
                        c++;
                }
        }
        printf("\nMemory block assigned:-\n ");
        printf("   Process\tSize\tBlock\n");
        for(int i=0;i<n;i++)
        {

                printf("\t%d\t%d\t",i+1,ps[i]);
                if(ab[i]!=-1)
                        printf("%d\n",ab[i]+1);
                else
                        printf("%s\n","Not allocated");
        }

}

int main()
{
        int m,bs[50],n,ps[50];
        printf("NEXT FIT ALGORITHM:\n");
        printf("\nEnter no of blocks: ");
        scanf("%d",&m);
```

```
46          printf("Enter each block size: ");
47          for(int i=0;i<m;i++)
48          scanf("%d",&bs[i]);
49          printf("Enter no of processes: ");
50          scanf("%d",&n);
51          printf("Enter each process size: ");
52          for(int i=0;i<n;i++)
53          scanf("%d",&ps[i]);
54          NextFit(bs,m,ps,n);
55 }
56
```

OUTPUT:

```
root@kali:~/maheysh/18_05_21LAB# gcc NEXTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
NEXT FIT ALGORITHM:

Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
Enter each process size: 212 417 112 426

Memory block assigned:-
    Process     Size    Block
        1       212     2
        2       417     5
        3       112     2
        4       426     Not allocated
root@kali:~/maheysh/18_05_21LAB#
```

BEST FIT ALGORITHM:

```c
#include<stdio.h>


void BestFit(int bs[50],int m,int ps[50],int n)
{
        int ab[50],a;
        for(int i=0;i<n;i++)
        {

                ab[i]=-1;
                for(int j=0;j<m;j++)
                        if(ps[i]<=bs[j])
                        {
                                a=bs[j]-ps[i];
                                if(ab[i]==-1)
                                        ab[i]=j;
                                else if(bs[ab[i]]-ps[i]>a)
                                        ab[i]=j;
                        }
                if(ab[i]!=-1)
                        bs[ab[i]]-=ps[i];
        }
        printf("\nMemory block assigned:-\n ");
        printf("   Process\tSize\tBlock\n");
        for(int i=0;i<n;i++)
        {

                printf("\t%d\t%d\t",i+1,ps[i]);
                if(ab[i]!=-1)
                        printf("%d\n",ab[i]+1);
                else
                        printf("%s\n","Not allocated");
        }

}

int main()
{
        int m,bs[50],n,ps[50];
        printf("BEST FIT ALGORITHM:\n");
        printf("\nEnter no of blocks: ");
        scanf("%d",&m);
        printf("Enter each block size: ");
        for(int i=0;i<m;i++)
        scanf("%d",&bs[i]);
        printf("Enter no of processes: ");
        scanf("%d",&n);
        printf("Enter each process size: ");
        for(int i=0;i<n;i++)
        scanf("%d",&ps[i]);
        BestFit(bs,m,ps,n);
}

```

OUTPUT:

```
root@kali:~/maheysh/18_05_21LAB# gcc BESTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
BEST FIT ALGORITHM:

Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
ter each process size: 212 417 112 426

Memory block assigned:-
    Process     Size    Block
        1       212     4
        2       417     2
        3       112     3
        4       426     5
root@kali:~/maheysh/18_05_21LAB#
```

WORST FIT ALGORITHM:

```c
#include<stdio.h>


void WorstFit(int bs[50],int m,int ps[50],int n)
{
        int ab[50],a;
        for(int i=0;i<n;i++)
        {

                ab[i]=-1;
                for(int j=0;j<m;j++)
                {

                        if(ps[i]<=bs[j])
                        {
                                a=bs[j]-ps[i];
                                if(ab[i]==-1)
                                        ab[i]=j;
                                else
                                {
                                        if(bs[ab[i]]-ps[i]<a)
                                        {
                                                ab[i]=j;
                                        }
                                }
                        }
                }
                if(ab[i]!=-1)
                        bs[ab[i]]-=ps[i];
        }
        printf("\nMemory block assigned:-\n ");
        printf("   Process\tSize\tBlock\n");
        for(int i=0;i<n;i++)
        {

                printf("\t%d\t%d\t",i+1,ps[i]);
                if(ab[i]!=-1)
                        printf("%d\n",ab[i]+1);
                else
                        printf("%s\n","Not allocated");
        }

}

```

```c
int main()
{
        int m,bs[50],n,ps[50];
        printf("WORST FIT ALGORITHM:\n");
        printf("\nEnter no of blocks: ");
        scanf("%d",&m);
        printf("Enter each block size: ");
        for(int i=0;i<m;i++)
        scanf("%d",&bs[i]);
        printf("Enter no of processes: ");
        scanf("%d",&n);
        printf("Enter each process size: ");
        for(int i=0;i<n;i++)
        scanf("%d",&ps[i]);
        WorstFit(bs,m,ps,n);
}

```

OUTPUT:

```
root@kali:~/maheysh/18_05_21LAB# gcc WORSTFIT_ALGO.c
root@kali:~/maheysh/18_05_21LAB# ./a.out
WORST FIT ALGORITHM:

Enter no of blocks: 5
Enter each block size: 100 500 200 300 600
Enter no of processes: 4
Enter each process size: 212 417 112 426

Memory block assigned:-
    Process     Size    Block
        1       212     5
        2       417     2
        3       112     5
        4       426     Not allocated
root@kali:~/maheysh/18_05_21LAB#
```

Result:

Various memory management algorithms were applied.Best fit algorithm was found out to be the most efficient.