

MEMORY ALLOCATION METHODS FOR FIXED PARTITION

PROGRAM

FIRST FIT

```
#include<stdio.h>
int main()
{
    int n,m,i,j;
    printf ("enter the number of process");
    scanf("%d",&n);
    printf ("enter the number of blocks");
    scanf("%d",&m);
    int p[n],b[m],allocation[n],a[m];
    for(i = 0; i < n; i++)
    {
        allocation[i] = -1;
    }
    for(i=0;i<n;i++)
    {
        printf("Process %d",i);
        printf("enter the process size");
        scanf("%d",&p[i]);
    }
    for(i=0;i<m;i++)
    {
        printf("Block %d",i);
        printf("enter the Block size");
        scanf("%d",&b[i]);
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            if ( p[i]<=b[j])
            {
                allocation[i] = j;
                a[i]=b[j];
                b[j] = b[j]-p[i];
                break;
            }
        }
    }
}
```

```

printf("\nProcess No.\tProcess Size\tBlock size.\tBlock num\n");
for (int i = 0; i < n; i++)
{
    printf(" %d\t\t", i);
    printf("%d\t", p[i]);

    if (allocation[i] != -1)
    {
        printf("%d \t", a[i]);
        printf("%d", allocation[i ]);
    }
    else
    {
        printf("Not Allocated");
    }
    printf("\n");
}
}

```

OUTPUT

```

lab1@sjcet-H81M-S:~/Allwina$ gcc ma.c
lab1@sjcet-H81M-S:~/Allwina$ ./a.out
enter the number of process4
enter the number of blocks5
Process 0enter the process size25
Process 1enter the process size30
Process 2enter the process size45
Process 3enter the process size20
Block 0enter the Block size50
Block 1enter the Block size40
Block 2enter the Block size30
Block 3enter the Block size80
Block 4enter the Block size20

Process No.      Process Size      Block size.      Block num
0                25                50               0
1                30                40               1
2                45                80               3
3                20                25               0
lab1@sjcet-H81M-S:~/Allwina$ 

```

BEST FIT

PROGRAM

```
#include<stdio.h>
int main()
{
    int n,m,i,j,pos,swap;
    printf ("enter the number of process");
    scanf("%d",&n);
    printf ("enter the number of blocks");
    scanf("%d",&m);
    int p[n],b[m],allocation[n],a[m],c[m];
    for(i = 0; i < n; i++)
    {
        allocation[i] = -1;
    }
    for(i=0;i<n;i++)
    {
        printf("Process %d",i);
        printf("enter the process size");
        scanf("%d",&p[i]);
    }
    for(i=0;i<m;i++)
    {
        printf("Block %d",i);
        printf("enter the Block size");
        scanf("%d",&b[i]);
    }
    //sorting
    for(i = 0; i < m- 1; i++)
    {
        pos=i;
        for(j = i + 1; j < m; j++)
        {
            if(b[pos] > b[j])
            pos=j;
        }
        if(pos != i)
        {
            swap=b[i];
            b[i]=b[pos];
            b[pos]=swap;
        }
    }
}
```

```

}
for (i = 0; i < n; i++)    {
    for (j = 0; j < m; j++)    {
        if ( p[i]<=b[j]) {
            allocation[i] = j;
            a[i]=b[j];
            b[j] = b[j]-p[i];
            break;
        }
    }
}
printf("\nProcess No.\tProcess Size\tBlock size.\n");
for (int i = 0; i < n; i++)
{
    printf(" %d\t\t", i);
    printf("%d\t", p[i]);

    if (allocation[i] != -1)
    {
        printf("%d \t",a[i]);    }
    else
    {
        printf("Not Allocated");
    }
    printf("\n");
}
}

```

OUTPUT

```

lab1@sjcet-H81M-S:~/Allwina$ ./a.out
enter the number of process4
enter the number of blocks5
Process 0enter the process size212
Process 1enter the process size417
Process 2enter the process size112
Process 3enter the process size426
Block 0enter the Block size100
Block 1enter the Block size500
Block 2enter the Block size200
Block 3enter the Block size300
Block 4enter the Block size600

Process No.      Process Size      Block size.
0                212              300
1                417              500
2                112              200
3                426              600
lab1@sjcet-H81M-S:~/Allwina$ 

```

WORST FIT

PROGRAM

```
#include<stdio.h>
int main()
{
    int n,m,i,j,pos,swap;
    printf ("enter the number of process");
    scanf("%d",&n);
    printf ("enter the number of blocks");
    scanf("%d",&m);
    int p[n],b[m],allocation[n],a[m],c[m],d=0;;
    for(i = 0; i < n; i++)
    {
        allocation[i] = -1;
    }
    for(i=0;i<n;i++)
    {
        printf("Process %d",i);
        printf("enter the process size");
        scanf("%d",&p[i]);
    }
    for(i=0;i<m;i++)
    {
        printf("Block %d",i);
        printf("enter the Block size");
        scanf("%d",&b[i]);
    }
    //sorting
    for(i = 0; i < m- 1; i++)
    {
        pos=i;
        for(j = i + 1; j < m; j++)
        {
            if(b[pos] > b[j])
            pos=j;
        }
        if(pos != i)
        {
            swap=b[i];
            b[i]=b[pos];
            b[pos]=swap;
        }
    }
    for(i = m-1; i >=0; i--)
    {
        c[d]=b[i];
        d=d+1;
    }
}
```

```

for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        if ( p[i]<=c[j])
        {
            allocation[i] = j;
            a[i]=c[j];
            c[j] = c[j]-p[i];
            break;
        }
    }
}
printf("\nProcess No.\tProcess Size\tBlock size.\n");
for (int i = 0; i < n; i++)
{
    printf(" %d\t\t\t", i);
    printf("%d\t\t", p[i]);

    if (allocation[i] != -1)
    {
        printf("%d \t",a[i]);
        //printf("%d", allocation[i ]);
    }
    else
    {
        printf("Not Allocated");
    }
    printf("\n");
}
}

```

OUTPUT

```

lab1@sjcet-H81M-S:~/Allwina$ gcc wf.c
lab1@sjcet-H81M-S:~/Allwina$ ./a.out
enter the number of process2
enter the number of blocks3
Process 0enter the process size1
Process 1enter the process size4
Block 0enter the Block size5
Block 1enter the Block size2
Block 2enter the Block size7

Process No.      Process Size      Block size.
0                  1                  7
1                  4                  6
lab1@sjcet-H81M-S:~/Allwina$ 

```

PAGE REPLACEMENT ALGORITHMS

FIFO

PROGRAM

```
#include<stdio.h>
int main()
{
    int n,i,f,j,avail,count=0,k;
    printf("enter the number of pages\n");
    scanf("%d",&n);
    int a[n],frame[10];
    printf("enter the reference string\n");
    for(i=0;i<n;i++)
    {
        scanf("\n%d",&a[i]);
    }
    printf(" enter the number of frames");
    scanf("\n%d",&f);
    for(i=0;i<f;i++)
        frame[i]= -1;
        j=0;
        printf("\ntref string\t page frames\n");
        for(i=0;i<n;i++)
        {
            printf("%d\t\t",a[i]);
            avail=0;
            for(k=0;k<f;k++)
                if(frame[k]==a[i])
                    avail=1;
            if (avail==0)
            {
                frame[j]=a[i];
                j=(j+1)%f;
                count++;
                for(k=0;k<f;k++)
                    printf("%d\t",frame[k]);
            }

            printf("\n");
        }

        printf("Page Fault Is %d",count);
        return 0;
    }
```

OUTPUT

```
student@student-Veriton-M200-H410:~/Allwina$ ./a.out
enter the number of pages
7
enter the reference string
1
3
0
3
5
6
3
enter the number of frames3
      ref string      page frames
1          1         -1      -1
3          1          3      -1
0          1          3       0
3
5          5          3       0
6          5          6       0
3          5          6       3
student@student-Veriton-M200-H410:~/Allwina$
```


LRU

PROGRAM

```
#include<stdio.h>
int main()
{
int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];
printf("Enter no of pages:");
scanf("%d",&n);
printf("Enter the reference string:");
for(i=0;i<n;i++)
scanf("\n%d",&p[i]);
printf("Enter no of frames:");
scanf("%d",&f);
q[k]=p[k];
printf("\n\t%d\n",q[k]);
c++;
k++;
for(i=1;i<n;i++)
{
    c1=0;
    for(j=0;j<f;j++)
    {
        if(p[i]!=q[j])
            c1++;
    }

    if(c1==f)
    {
        c++;
        if(k<f)
        {
            q[k]=p[i];
            k++;
            for(j=0;j<k;j++)
                printf("\t%d",q[j]);
            printf("\n");
        }
        else
        {
            for(r=0;r<f;r++)
            {
                c2[r]=0;
                for(j=i-1;j<n;j--)
                {
                    if(q[r]!=p[j])
                        c2[r]++;
                    else
                        break;
                }
            }
        }
    }
}
```

```

        }
    }
    for(r=0;r<f;r++)
        b[r]=c2[r];
    for(r=0;r<f;r++)
    {
        for(j=r;j<f;j++)
        {
            if(b[r]<b[j])
            {
                t=b[r];
                b[r]=b[j];
                b[j]=t;
            }
        }
    }
    for(r=0;r<f;r++)
    {
        if(c2[r]==b[0])
            q[r]=p[i];
        printf("\t%d",q[r]);
    }
    printf("\n");
}

}

printf("\nThe no of page faults is %d",c);
}

```

OUTPUT

```

student@student-Veriton-M200-H410:~/Allwina$ gcc lru.c
student@student-Veriton-M200-H410:~/Allwina$ ./a.out
Enter no of pages:10
Enter the reference string:4
7
6
1
7
6
1
2
7
2
Enter no of frames:3
4
4      7
4      7      6
1      7      6
1      2      6
1      2      7
The no of page faults is 6student@student-Veriton-M200-H410:~/Allwina$ 

```

