CSE2008 (Operating Systems) Lab-7

KHAN MOHD OWAIS RAZA
20BCD7138

Q.1 Consider memory allocation strategy using Fixed partitioning. Given memory partition sizes and process sizes write a program to calculate total internal fragmentation using

1. First Fit
2. Best Fit
3. Worst Fit strategy

Java Code & Output :-

```
/* KHAN MOHD OWAIS RAZA */
/* 20BCD7138*/
/* CSE2008 (OPERATING SYSTEMS) LAB PRACTICAL-7 */
/* Consider memory allocation strategy using Fixed partitioning.
   Given memory partition sizes and process sizes write a program
   to calculate total internal fragmentation using
   (1) first fit
   (2) best fit
   (3) worst fit
*/
package owaisraza.CSE2008_Lab7;
import java.util.Scanner;
public class MyClass {
public static void FIRST_FIT(int PARTITION[],int PARTITION_LENGTH,int
PROCESS[],int PROCESS_LENGTH) {
boolean VISITED[] = new boolean[PARTITION_LENGTH];
for (int A = 0; A < PARTITION_LENGTH; A++) {
VISITED[A] = false;
}
System.out.println(" ");
System.out.println("(1) FIRST FIT");
for (int A = 0; A < PROCESS_LENGTH; A++) {
boolean ALLOCATE = false;
for (int B = 0; B < PARTITION_LENGTH; B++) {
if(PARTITION[B] > PROCESS[A] && VISITED[B] == false) {
VISITED[B] = true;
ALLOCATE = true;
System.out.format("PROCESS %d IS ALLOCATED TO PARTITION %d\n", A+1, B+1);
break;
}}
if(!ALLOCATE) {
System.out.format("PROCESS %d IS NOT ALLOCATED BY ANY PARTITION\n", A+1);
}}
}
public static void BEST_FIT(int PARTITION[], int PARTITION_LENGTH, int
PROCESS[], int PROCESS_LENGTH) {
boolean VISITED[] = new boolean[PARTITION_LENGTH];
```

```
for (int A = 0; A < PARTITION_LENGTH; A++) {
VISITED[A] = false;
}
System.out.println(" ");
System.out.println("(2) BEST FIT ");
for (int A = 0; A < PROCESS_LENGTH; A++) {
int X = -1;
for (int B = 0; B < PARTITION_LENGTH; B++) {
if(PARTITION[B] > PROCESS[A] && VISITED[B] == false) {
if(X == -1 || (PARTITION[B] - PROCESS[A]) < (PARTITION[X] - PROCESS[A])) {
X = B;
}}
}
if(X == -1) {
System.out.format("PROCESS %d IS NOT ALLOCATED BY ANY PARTITION\n", A+1);
}
else {
System.out.format("PROCESS %d IS ALLOCATED TO PARTITION %d\n", A+1, X+1);
VISITED[X] = true;
}}
}
public static void WORST_FIT(int PARTITION[], int PARTITION_LENGTH, int
PROCESS[], int PROCESS_LENGTH) {
boolean visited[] = new boolean[PARTITION_LENGTH];
for (int A = 0; A < PARTITION_LENGTH; A++) {
visited[A] = false;
}
System.out.println(" ");
System.out.println("(3) WORST FIT");
for (int A = 0; A < PROCESS_LENGTH; A++) {
int X = -1;
for (int B = 0; B < PARTITION_LENGTH; B++) {
if(PARTITION[B] > PROCESS[A] && visited[B] == false) {
if(X == -1 || (PARTITION[B] - PROCESS[A]) > (PARTITION[X] - PROCESS[A])) {
X = B;
}}
}
if(X == -1) {
System.out.format("PROCESS %d IS NOT ALLOCATED BY ANY PARTITION\n", A+1);
}
else {
System.out.format("PROCESS %d IS ALLOCATED TO PARTITION %d\n", A+1, X+1);
visited[X] = true;
}}
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("ENTER NO. OF MEMORY PARTITIONS: ");
int PARTITION_LENGTH = sc.nextInt();
int PARTITION[] = new int[PARTITION_LENGTH];
for (int A = 0; A < PARTITION_LENGTH; A++) {
System.out.print("ENTER SIZE OF PARTITION " + (A+1) +" :");
PARTITION[A] = sc.nextInt();
}
```

```
System.out.print("ENTER NO. OF PROCESSES: ");
int PROCESS_LENGTH = sc.nextInt();
int PROCESS[] = new int[PROCESS_LENGTH];
for (int A = 0; A < PROCESS_LENGTH; A++) {
System.out.print("ENTER SIZE OF PROCESS " + (A+1)+" :");
PROCESS[A] = sc.nextInt();
}
FIRST_FIT(PARTITION, PARTITION_LENGTH, PROCESS, PROCESS_LENGTH);
BEST_FIT(PARTITION, PARTITION_LENGTH, PROCESS, PROCESS_LENGTH);
WORST_FIT(PARTITION, PARTITION_LENGTH, PROCESS, PROCESS_LENGTH);
}}
```

```
<terminated> MyClass (4) [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe
ENTER NO. OF MEMORY PARTITIONS: 5
ENTER SIZE OF PARTITION 1 :100
ENTER SIZE OF PARTITION 2 :500
ENTER SIZE OF PARTITION 3 :200
ENTER SIZE OF PARTITION 4 :300
ENTER SIZE OF PARTITION 5 :600
ENTER NO. OF PROCESSES: 4
ENTER SIZE OF PROCESS 1 :212
ENTER SIZE OF PROCESS 2 :417
ENTER SIZE OF PROCESS 3 :112
ENTER SIZE OF PROCESS 4 :426

(1) FIRST FIT
PROCESS 1 IS ALLOCATED TO PARTITION 2
PROCESS 2 IS ALLOCATED TO PARTITION 5
PROCESS 3 IS ALLOCATED TO PARTITION 3
PROCESS 4 IS NOT ALLOCATED BY ANY PARTITION

(2) BEST FIT
PROCESS 1 IS ALLOCATED TO PARTITION 4
PROCESS 2 IS ALLOCATED TO PARTITION 2
PROCESS 3 IS ALLOCATED TO PARTITION 3
PROCESS 4 IS ALLOCATED TO PARTITION 5

(3) WORST FIT
PROCESS 1 IS ALLOCATED TO PARTITION 5
PROCESS 2 IS ALLOCATED TO PARTITION 2
PROCESS 3 IS ALLOCATED TO PARTITION 4
PROCESS 4 IS NOT ALLOCATED BY ANY PARTITION
```