Name : Khune Radha Rahul

Class : TE Comp-2

Roll no : 07(F22112010)

**Best fit**

**Input:**

def bestFit(blockSize, m, processSize, n):

allocation = [-1] \* n

for i in range(n):

bestIdx = -1

for j in range(m):

if blockSize[j] >= processSize[i]:

if bestIdx == -1:

bestIdx = j

elif blockSize[bestIdx] > blockSize[j]:

bestIdx = j

if bestIdx != -1:

allocation[i] = bestIdx

blockSize[bestIdx] -= processSize[i]

print("Process No. Process Size Block no.")

for i in range(n):

print(i + 1, " ", processSize[i],

end=" ")

if allocation[i] != -1:

print(allocation[i] + 1)

else:

print("Not Allocated")

def main():

blockSize = [100, 500, 200, 300, 600]

processSize = [212, 417, 112, 426]

m = len(blockSize)

n = len(processSize)

bestFit(blockSize, m, processSize, n)

main()

**Output:**

admin1@linux:~$ python3 b.py

Enter the number of memory blocks: 3

Enter size of block 1: 2

Enter size of block 2: 6

Enter size of block 3: 9

Enter the number of processes: 3

Enter size of process 1: 2

Enter size of process 2: 5

Enter size of process 3: 9

Process No. Process Size Block no.

1 2 1

2 5 2

3 9 3

**First fit**

**Input:**

def firstFit(blockSize, m, processSize, n):

allocation = [-1] \* n

for i in range(n):

for j in range(m):

if blockSize[j] >= processSize[i]:

allocation[i] = j

blockSize[j] -= processSize[i]

break

print("Process No. Process Size Block no.")

for i in range(n):

print(i + 1, " ", processSize[i], end=" ")

if allocation[i] != -1:

print(allocation[i] + 1)

else:

print("Not Allocated")

def main():

m = int(input("Enter the number of memory blocks: "))

blockSize = []

for i in range(m):

size = int(input(f"Enter size of block {i + 1}: "))

blockSize.append(size)

n = int(input("Enter the number of processes: "))

processSize = []

for i in range(n):

size = int(input(f"Enter size of process {i + 1}: "))

processSize.append(size)

firstFit(blockSize, m, processSize, n)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Output:**

admin1@linux:~$ python3 f.py

Enter the number of memory blocks: 4

Enter size of block 1: 8

Enter size of block 2: 7

Enter size of block 3: 5

Enter size of block 4: 9

Enter the number of processes: 4

Enter size of process 1: 3

Enter size of process 2: 5

Enter size of process 3: 6

Enter size of process 4: 1

Process No. Process Size Block no.

1 3 1

2 5 1

3 6 2

4 1 2

**Next fit**

**Input:**

def NextFit(blockSize, m, processSize, n):

allocation = [-1] \* n

j = 0

t = m - 1

for i in range(n):

while j < m:

if blockSize[j] >= processSize[i]:

allocation[i] = j

blockSize[j] -= processSize[i]

t = (j - 1) % m

break

if t == j:

t = (j - 1) % m

break

j = (j + 1) % m

print("Process No. Process Size Block no.")

for i in range(n):

print("\t", i + 1, "\t\t\t", processSize[i], end="\t\t\t")

if allocation[i] != -1:

print(allocation[i] + 1)

else:

print("Not Allocated")

def main():

blockSize = [5, 10, 20]

processSize = [10, 20, 5]

m = len(blockSize)

n = len(processSize)

NextFit(blockSize, m, processSize, n)

main()

**Output:**

admin1@linux:~$ python3 n.py

Enter the number of memory blocks: 4

Enter size of block 1: 2

Enter size of block 2: 7

Enter size of block 3: 4

Enter size of block 4: 6

Enter the number of processes: 4

Enter size of process 1: 1

Enter size of process 2: 9

Enter size of process 3: 6

Enter size of process 4: 4

Process No. Process Size Block no.

1 1 1

2 9 Not Allocated

3 6 2

4 4 3

**Worst fit**

**Input:**

def worstFit(blockSize, m, processSize, n):

allocation = [-1] \* n

for i in range(n):

wstIdx = -1

for j in range(m):

if blockSize[j] >= processSize[i]:

if wstIdx == -1:

wstIdx = j

elif blockSize[wstIdx] < blockSize[j]:

wstIdx = j

if wstIdx != -1:

allocation[i] = wstIdx

blockSize[wstIdx] -= processSize[i]

print("Process No. Process Size Block no.")

for i in range(n):

print(i + 1, " ",

processSize[i], end=" ")

if allocation[i] != -1:

print(allocation[i] + 1)

else:

print("Not Allocated")

def main():

blockSize = [100, 500, 200, 300, 600]

processSize = [212, 417, 112, 426]

m = len(blockSize)

n = len(processSize)

worstFit(blockSize, m, processSize, n)

main()

**Output:**

admin1@linux:~$ python3 w.py

Enter the number of memory blocks: 5

Enter size of block 1: 3

Enter size of block 2: 8

Enter size of block 3: 6

Enter size of block 4: 9

Enter size of block 5: 1

Enter the number of processes: 5

Enter size of process 1: 2

Enter size of process 2: 8

Enter size of process 3: 5

Enter size of process 4: 9

Enter size of process 5: 1

Process No. Process Size Block no.

1 2 4

2 8 2

3 5 4

4 9 Not Allocated

5 1 3