|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap  Id: | 60004220127 |
| Class: | T. Y. B.Tech (Computer Engineering) |
| Course: | Processor Organization and Architecture (POA) |
| Course Code: | DJ19CEL502 |
| Experiment  No.: | 04 |

**AIM:** Memory allocation techniques like first fit, best fit, worst fit and next fit.

FIRST FIT:

**CODE**:

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")

if \_\_name\_\_ == '\_\_main\_\_':

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

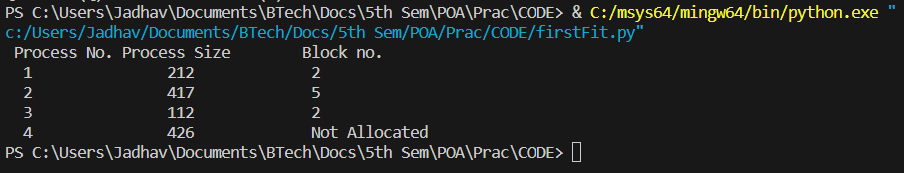
processSize = [212, 417, 112, 426]

m = len(blockSize)

n = len(processSize)

firstFit(blockSize, m, processSize, n)

**OUTPUT**:



BEST FIT:

**CODE**:

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")

# Driver code

if \_\_name\_\_ == '\_\_main\_\_':

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

processSize = [212, 417, 112, 426]

m = len(blockSize)

n = len(processSize)

bestFit(blockSize, m, processSize, n)

**OUTPUT**:

A screen shot of a computer

Description automatically generated

WORST FIT:

**CODE**:

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")

if \_\_name\_\_ == '\_\_main\_\_':

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

processSize = [212, 417, 112, 426]

m = len(blockSize)

n = len(processSize)

worstFit(blockSize, m, processSize, n)

**OUTPUT**:

A screen shot of a computer

Description automatically generated

NEXT FIT:

**CODE:**

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")

if \_\_name\_\_ == '\_\_main\_\_':

blockSize = [5, 10, 20]

processSize = [10, 20, 5]

m = len(blockSize)

n = len(processSize)

NextFit(blockSize, m, processSize, n)

**OUTPUT:**

**A screen shot of a computer

Description automatically generated**