**Expr 10 a: Best Fit**

**Best Fit code:**

class Block:

def \_\_init\_\_(self, size, num):

self.size = size

self.num = num

self.allocated = False

class Process:

def \_\_init\_\_(self, size, num):

self.size = size

self.num = num

self.allocated = None

def best\_fit(blocks, processes):

print(&quot;Process No.\tProcess Size\tBlock No.&quot;)

for p in processes:

best = None

for b in blocks:

if not b.allocated and b.size &gt;= p.size:

if best is None or b.size &lt; best.size:

best = b

if best:

best.allocated = True

p.allocated = best.num

print(f&quot;{p.num}\t\t{p.size}\t\t{p.allocated if p.allocated else &#39;Not Allocated&#39;}&quot;)

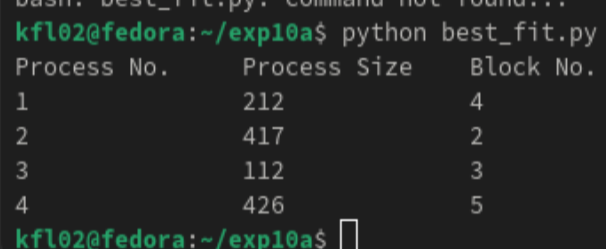
# Test data

blocks = [Block(s, i) for i, s in enumerate([100, 500, 200, 300, 600], 1)]

processes = [Process(s, i) for i, s in enumerate([212, 417, 112, 426], 1)]

best\_fit(blocks, processes)

**Output:**



**Result:**

Thus the Best fit Code is implemented in fedora using the fedora language