

1. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

#### FIRST-FIT

|     |     |     |     |     |     |     |     |     |    |     |  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|--|
| 300 |     | 600 |     | 350 |     | 200 |     | 750 |    | 125 |  |
| 115 | 185 | 600 |     | 350 |     | 200 |     | 750 |    | 125 |  |
| 115 | 185 | 500 | 100 | 350 |     | 200 |     | 750 |    | 125 |  |
| 115 | 185 | 500 | 100 | 350 |     | 200 | 358 | 392 |    | 125 |  |
| 115 | 185 | 500 | 100 | 200 | 150 | 200 | 358 | 392 |    | 125 |  |
| 115 | 185 | 500 | 100 | 200 | 150 | 200 | 358 | 375 | 17 | 125 |  |

#### BEST-FIT

|     |  |     |     |     |     |     |     |     |     |     |  |
|-----|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|
| 300 |  | 600 |     | 350 |     | 200 |     | 750 |     | 125 |  |
| 300 |  | 600 |     | 350 |     | 200 |     | 750 | 115 | 10  |  |
| 300 |  | 500 | 100 | 350 |     | 200 |     | 750 | 115 | 10  |  |
| 300 |  | 500 | 100 | 350 |     | 200 | 358 | 392 | 115 | 10  |  |
| 300 |  | 500 | 100 | 350 | 200 | 358 | 392 | 115 | 10  |     |  |
| 300 |  | 500 | 100 | 350 | 200 | 358 | 375 | 17  | 115 | 10  |  |

#### WORST-FIT

|     |  |     |     |     |     |     |     |     |     |     |  |
|-----|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|
| 300 |  | 600 |     | 350 |     | 200 |     | 750 |     | 125 |  |
| 300 |  | 600 |     | 350 |     | 200 | 115 | 635 |     | 125 |  |
| 300 |  | 600 |     | 350 |     | 200 | 115 | 500 | 135 | 125 |  |
| 300 |  | 358 | 242 | 350 |     | 200 | 115 | 500 | 135 | 125 |  |
| 300 |  | 358 | 242 | 200 | 150 | 200 | 115 | 500 | 135 | 125 |  |

Can't place process of size 375KB

In term of efficiency, we can rank 3 algorithms as: Best-fit > First-fit > Worst-fit.

2. Student write a short report that compares the advantages as well as disadvantages of the allocation algorithms, namely First-Fit, Best-Fit, Worst-Fit.

## First Fit

In the first fit approach is to allocate the first free partition or hole large enough which can accommodate the process. It finishes after finding the first suitable free partition.

### Advantage

Fastest algorithm because it searches as little as possible

### Disadvantage

The remaining unused memory areas left after allocation become waste if it is too small. Thus request for larger memory requirement cannot be accomplished.

## Best Fit

The best fit deals with allocating the smallest free partition which meets the requirement of the requesting process. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is close to actual process size needed.

### Advantage

Memory utilization is much better than first fit as it searches the smallest free partition first available. It tend to leave very large holes and very small holes.

### Disadvantage

It is slower and may even tend to fill up memory with tiny useless holes and wasted memory space. We have to search and find a space whose size is either similar to file size/process size or is slightly more. Another problem associated with this technique is that if we want to extend the file contents than it can't be implemented.

## Worst fit

In worst fit approach is to locate largest available free portion so that the portion left will be big enough to be useful. It is the reverse of best fit.

**Advantage**

Reduces the rate of production of small gaps. Also, if a program wants to increase its content, it can be easily implemented. Since the gaps is fairly large, these gaps can be useful to allocate other process.

**Disadvantage**

If a process requiring larger memory arrives at a later stage then it cannot be accommodated as the largest hole is already split and occupied. It also has to search the entire list to find the biggest hole, which take time and thus, slow. Also, simulation show that worst-fit is worst in terms of storage utilization.