**Week 3 Assignment**

**Analysis of Dynamic Storage Allocation Algorithms**

Eric Vara

The University of Arizona Global Campus

CPT 304: Operating Systems Theory and Design

Professor Joshua Reichard

January 15, 2023

**Introduction**

This paper provides an analysis of three dynamic storage allocation algorithms: first-fit, best-fit, and worst-fit. Using a scenario involving eight unique processes and ten memory partitions, the paper will illustrate how each algorithm allocates memory and discuss the efficiency and execution speed of these models.

**Scenario Description**

The given scenario involves allocating eight processes with sizes (in KB) of 21, 77, 199, 380, 489, 212, 139, and 302 into memory partitions sized (in KB) at 150, 275, 425, 300, 80, 117, 500, 35, 313, and 269, respectively.

**First-Fit Algorithm**

**21 KB Process:** Placed in the first partition of 150 KB.

Remaining partitions: 150 (129 KB remaining), 275, 425, 300, 80, 117, 500, 35, 313, 269.

**77 KB Process:** Placed in the first partition (129 KB remaining).

Remaining partitions: 150 (52 KB remaining), 275, 425, 300, 80, 117, 500, 35, 313, 269.

**199 KB Process:** Placed in the second partition of 275 KB.

Remaining partitions: 150 (52 KB remaining), 275 (76 KB remaining), 425, 300, 80, 117, 500, 35, 313, 269.

**380 KB Process:** Placed in the fourth partition of 300 KB.

Remaining partitions: 150 (52 KB remaining), 275 (76 KB remaining), 425, 300 (not enough space), 80, 117, 500, 35, 313, 269.

489 KB Process: Placed in the seventh partition of 500 KB.

Remaining partitions: 150 (52 KB remaining), 275 (76 KB remaining), 425, 300 (not enough space), 80, 117, 500 (11 KB remaining), 35, 313, 269.

**212 KB Process:** Placed in the third partition of 425 KB.

Remaining partitions: 150 (52 KB remaining), 275 (76 KB remaining), 425 (213 KB remaining), 300 (not enough space), 80, 117, 500 (11 KB remaining), 35, 313, 269.

**139 KB Process:** Placed in the third partition (213 KB remaining).

Remaining partitions: 150 (52 KB remaining), 275 (76 KB remaining), 425 (74 KB remaining), 300 (not enough space), 80, 117, 500 (11 KB remaining), 35, 313, 269.

**302 KB Process:** Placed in the ninth partition of 313 KB.

Remaining partitions: 150 (52 KB remaining), 275 (76 KB remaining), 425 (74 KB remaining), 300 (not enough space), 80, 117, 500 (11 KB remaining), 35, 313 (11 KB remaining), 269.

**Best-Fit Algorithm**

**21 KB Process:** Placed in the eighth partition of 35 KB.

Remaining partitions: 150, 275, 425, 300, 80, 117, 500, 35 (14 KB remaining), 313, 269.

**77 KB Process:** Placed in the sixth partition of 117 KB.

Remaining partitions: 150, 275, 425, 300, 80, 117 (40 KB remaining), 500, 35 (14 KB remaining), 313, 269.

**199 KB Process:** Placed in the tenth partition of 269 KB.

Remaining partitions: 150, 275, 425, 300, 80, 117 (40 KB remaining), 500, 35 (14 KB remaining), 313, 269 (70 KB remaining).

**380 KB Process:** Placed in the fourth partition of 300 KB.

Remaining partitions: 150, 275, 425, 300 (not enough space), 80, 117 (40 KB remaining), 500, 35 (14 KB remaining), 313, 269 (70 KB remaining).

**489 KB Process:** Placed in the seventh partition of 500 KB.

Remaining partitions: 150, 275, 425, 300 (not enough space), 80, 117 (40 KB remaining), 500 (11 KB remaining), 35 (14 KB remaining), 313, 269 (70 KB remaining).

**212 KB Process:** Placed in the third partition of 425 KB.

Remaining partitions: 150, 275, 425 (213 KB remaining), 300 (not enough space), 80, 117 (40 KB remaining), 500 (11 KB remaining), 35 (14 KB remaining), 313, 269 (70 KB remaining).

**139 KB Process:** Placed in the third partition (213 KB remaining).

Remaining partitions: 150, 275, 425 (74 KB remaining), 300 (not enough space), 80, 117 (40 KB remaining), 500 (11 KB remaining), 35 (14 KB remaining), 313, 269 (70 KB remaining).

**302 KB Process:** Placed in the second partition of 275 KB.

Remaining partitions: 150, 275 (not enough space), 425 (74 KB remaining), 300 (not enough space), 80, 117 (40 KB remaining), 500 (11 KB remaining), 35 (14 KB remaining), 313, 269 (70 KB remaining).

**Worst-Fit Algorithm**

**21 KB Process:** Placed in the seventh partition of 500 KB.

Remaining partitions: 150, 275, 425, 300, 80, 117, 500 (479 KB remaining), 35, 313, 269.

**77 KB Process:** Placed in the seventh partition (479 KB remaining).

Remaining partitions: 150, 275, 425, 300, 80, 117, 500 (402 KB remaining), 35, 313, 269.

**199 KB Process:** Placed in the seventh partition (402 KB remaining).

Remaining partitions: 150, 275, 425, 300, 80, 117, 500 (203 KB remaining), 35, 313, 269.

**380 KB Process:** Placed in the third partition of 425 KB.

Remaining partitions: 150, 275, 425 (45 KB remaining), 300, 80, 117, 500 (203 KB remaining), 35, 313, 269.

**489 KB Process:** Cannot be placed as no partition is large enough.

Remaining partitions: 150, 275, 425 (45 KB remaining), 300, 80, 117, 500 (203 KB remaining), 35, 313, 269.

**212 KB Process:** Placed in the second partition of 275 KB.

Remaining partitions: 150, 275 (63 KB remaining), 425 (45 KB remaining), 300, 80, 117, 500 (203 KB remaining), 35, 313, 269.

**139 KB Process:** Placed in the seventh partition (203 KB remaining).

Remaining partitions: 150, 275 (63 KB remaining), 425 (45 KB remaining), 300, 80, 117, 500 (64 KB remaining), 35, 313, 269.

**302 KB Process:** Placed in the fourth partition of 300 KB.

Remaining partitions: 150, 275 (63 KB remaining), 425 (45 KB remaining), 300 (not enough space), 80, 117, 500 (64 KB remaining), 35, 313, 269.

**Visual Illustration of Each Algorithm**

**A diagram of a process

Description automatically generated**

**A diagram of a process

Description automatically generated**

**A diagram of a diagram

Description automatically generated**

**Visual Illustration of Each Algorithm**

**Efficiency in Memory Use:**

* Best-Fit tends to leave the smallest leftover partitions, making more efficient use of memory.
* First-Fit can result in larger unused memory spaces due to its sequential nature.
* Worst-Fit, by design, leaves the largest partitions free, which can be inefficient.

**Execution Speed:**

* First-Fit is generally the quickest as it places a process in the first available partition without searching for a better fit.
* Best-Fit requires more time to search the entire list of partitions to find the smallest adequate space.
* Worst-Fit also needs to search for the largest partition, which can be time-consuming.

**Conclusion**

In conclusion, while the best-fit algorithm generally makes the most efficient use of memory, its execution speed is slower compared to the first-fit algorithm. The worst-fit algorithm, while useful in certain contexts, often results in less efficient memory usage. The choice of algorithm depends on the specific requirements of the system, balancing the need for efficient memory use against the need for speed in allocation.

**Work Cited**

OpenGenus IQ. (n.d.). First, Best and Worst fit Strategies (Memory Allocation Strategies). OpenGenus. Retrieved from <https://iq.opengenus.org/first-best-and-worst-fit-strategies/>

PrepInsta. (2022, April 1). First Fit Best Fit Worst Fit in OS (Example). PrepInsta. Retrieved from <https://prepinsta.com/first-fit-best-fit-worst-fit-in-os-example/>