

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



HỆ ĐIỀU HÀNH (CO2018)

Trả lời câu hỏi:

Lab 8: Memory Allocation

Giảng viên hướng dẫn: La Hoàng Lộc.
Lớp: L05.
Sinh viên thực hiện: Đinh Như Tân - 1915040.

HỒ CHÍ MINH, THÁNG 5 NĂM 2021



Contents

1	Exercise 1:	2
2	Exercise 2:	2

1 Exercise 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.

Trả lời:

Size of process	First-Fit	Best-Fit	Worst-Fit
115	300	125	750
500	600	600	600
358	750	750	X
200	350	200	350
375	X	X	X

- **First-Fit** sử dụng bộ nhớ chưa tối ưu, xảy ra hiện tượng phân mảnh nội khi các process yêu cầu vùng nhớ ít lại sử dụng phân vùng có kích thước lớn hơn, cụ thể là 115 KB, sử dụng 300 KB; 358 KB sử dụng 700 KB; 200 KB sử dụng 350 KB.
- **Best-Fit** sử dụng bộ nhớ tối ưu hơn **First-Fit**, tránh được hiện tượng phân mảnh nội, tìm các phân vùng đủ với các yêu cầu vùng nhớ, chẳng hạn 115 chỉ dùng 125; 200 sử dụng vừa đủ 200 KB.
- Tuy nhiên, so về tốc độ cấp phát phân vùng, thuật toán **Best-Fit** tìm kiếm chậm hơn so với **First-Fit** khi duyệt toàn bộ các phân vùng trong bộ nhớ để tìm phân vùng thích hợp nhất. Do đó cần tổ chức thuật toán để tối ưu khả năng tìm kiếm cho **Best-Fit**.
- Còn thuật toán **Worst-Fit** sử dụng vùng nhớ lãng phí, gây ra hiện tượng phân mảnh nội, đồng thời không đủ phân vùng trống cho các process yêu cầu vùng nhớ lớn hơn.

⇒ Dựa vào kết quả có như trên, có thể xếp hạng giải thuật theo thứ tự sử dụng hiệu quả bộ nhớ (memory) giảm dần lần lượt là: **Best-fit** → **First-fit** → **Worst-fit**.

2 Exercise 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**.

Trả lời:

	First-fit	Best-fit	Worst-fit
Advantages	Cấp phát nhanh, thời gian yêu cầu thực thi ngắn; Hiện thực đơn giản	Cấp phát vùng nhớ tối ưu, tránh hiện tượng phân mảnh nội, giảm thiểu lãng phí vùng nhớ	Nhanh, đơn giản và dễ hiện thực
Disadvantages	Sử dụng lãng phí vùng nhớ, không cấp phát vùng nhớ tối ưu, gây ra hiện tượng phân mảnh nội; Yêu cầu tổ chức phân vùng hợp lý	Tìm kiếm không nhanh, yêu cầu thời gian tìm kiếm phân vùng thích hợp.; Hiện thực phức tạp, cần tổ chức thuật toán, phân vùng thích hợp để tối ưu tốc độ tìm kiếm	Lãng phí vùng nhớ, tiêu tốn thời gian, yêu cầu tìm kiếm, không tối ưu, gây hiện tượng phân mảnh nội.