ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

Đại học Bách Khoa

Khoa Khoa học và Kĩ thuật Máy tính

HỆ ĐIỀU HÀNH
Báo cáo Bài tập lớn 2
SIMPLE OPERATING SYSTEM

Giảng viên hướng dẫn : TA. Trần Ngọc Anh Tú

Nhóm sinh viên thực hiện:

Lê Trọng Tuân	1613860
Trần Vũ Hiếu	1611060
Trần Thị Kim Ngân	1612176
Nguyễn Trọng Quý	1512748

CHỦ ĐỀ BÁO CÁO
SCHEDULING & MEMORY MANAGEMENT

MỤC LỤC

Contents

1. SCHEDULING

Question: What is the advantage of using priority feedback queue in comparison with other scheduling algorithms you have learned?

(Ưu điểm của priority feedback queue so với những giải thuật định thời khác?

+ Đảm bảo sự công bằng cho các process nhờ cách định thời tương tự Round Robin ở ready queue (mỗi process chỉ được phép thực thi trong một time slice).

+ Vẫn giữ được độ ưu tiên (process có độ ưu tiên cao được thực thi trước ở ready queue).

+ Tránh được starvation và thời gian đáp ứng ngắn (các process mới dù độ ưu tiên cao hay thấp đều được vào ready queue trước còn những process vừa thực thi xong ở ready queue sẽ phải xuống đợi ở run queue).

Cơ sở lý thuyết priority feedback queue :

Đối với mỗi chương trình mới, trình nạp sẽ tạo một process mới và gán một PCB mới cho nó. Trình nạp sau đó đọc và sao chép nội dung của chương trình vào text segment của quy trình mới (được chỉ ra bởi con trỏ mã trong PCB của process). Cuối cùng, PCB của quá trình được đẩy vào ready_queue và đợi CPU. CPU chạy các process theo kiểu vòng tròn. Mỗi process được phép chạy tối đa một khoảng thời gian nhất định. Sau đó, CPU buộc phải tạm dừng process và đẩy nó vào run_queue. CPU sau đó chọn lên một process khác từ ready_queue và tiếp tục chạy. Từ khi Cpu không lấy process quay trở lại reday_queue sau khi ngừng nó, ready_queue sẽ sớm hay muộn rỗng (số của quy trình là vô tận).. Nếu hiện tượng này xảy ra, the scheduler sẽ di chuyển tất cả các processes chờ đợi khi run-queue trở lại ready_queue để cho CPU tiếp tục chạy lại quá trình bị tạm dừng.

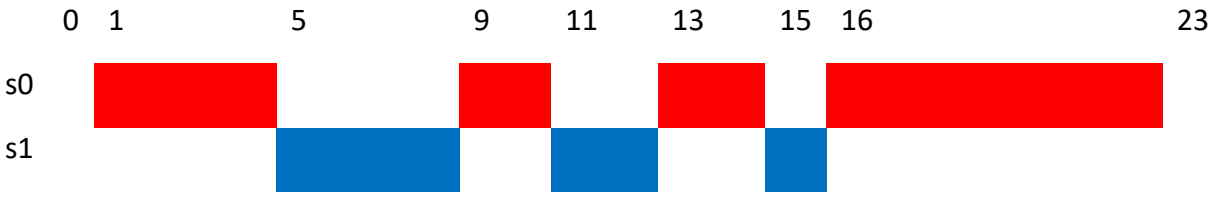
Scheduling Test 0

+ Quantum time = 2

+ 1 CPU

+ 2 process:

Process	**Arrival Time**	**Burst Time**	**Priority**
s0	0	15	12
s1	4	7	20



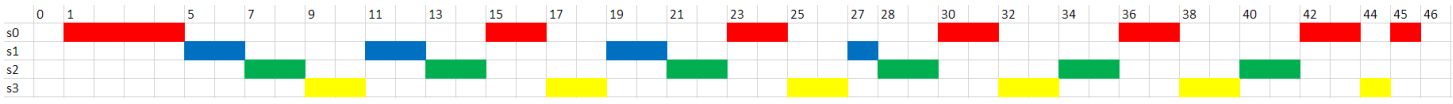
Scheduling Test 1

+ Quantum time = 2

+ 1 CPU

+ 4 process:

Process	Arrival Time	Burst Time	Priority
s0	0	15	12
s1	4	7	20
s2	6	12	20
s3	7	11	7



2.MEMORY MANAGEMENT

Question: What is the advantage and disadvantage of segmentation with paging ?

Ưu điểm:

+ Khắc phục được phân mảnh ngoại :paging đoạn,cho phép các page của đoạn được nạp vào các frame không cần nằm liên tục nhau.

+ Giảm bộ nhớ so với phân trang : vì khắc phục được phân mảnh ngoại.

+ Bảo vệ bộ nhớ : việc bảo vệ bộ nhớ được thực hiện bằng cách gắn với frame các bit bảo vệ (protection bit) được giữ trong bảng phân trang.

Nhược điểm:

+ Vẫn tồn tại phân mảnh nội .

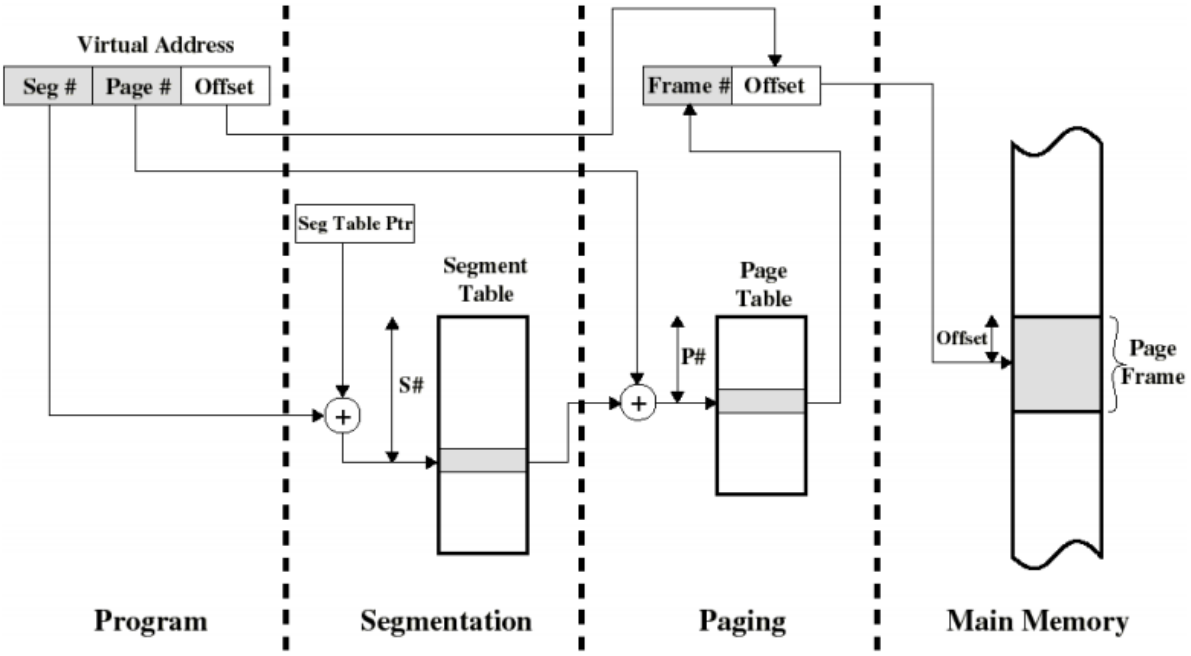
Cơ sở lý thuyết segmentation with paging:

- Một process sẽ có:

  +Một bảng phân đoạn.

  +Nhiều bảng phân trang : Mỗi đoạn có 1 bảng phân trang.

- Giải thuật :



+ Một địa chỉ ảo bao gồm : segment index, page index và offset (offset là phần bù của page).

+ Dùng segment index ánh xạ vào Segment Table để tìm Page Table cho segment index.

+ Dùng page index ánh xạ vào Page Table của segment index vừa tìm được để tìm page frame index trong Main Memory.

+ Cộng frame address với offset ta được physical address trong Main Memory.

- Cấp phát vùng nhớ cho process (alloc_mem) :

  + Tính số page cần cho process.

  + Kiểm tra số page free có đủ cho số page cần cấp cho process.

  + Nếu đủ page free :

    - Tìm size của Segment Table và size của Page Table của Process.

- Tìm các page free rồi thực hiện update các giá trị : next của page,process,index của page trong _mem_stat và update page table. Đến khi nào xong thì dừng.

Test

+ Nội dung Process m0:

```
1 7
alloc 13535 0
alloc 1568 1
free 0
alloc 1386 2
alloc 4564 4
write 102 1 20
write 21 2 1000
```

+ Nội dung process m1:

```
1 8
alloc 13535 0
alloc 1568 1
free 0
alloc 1386 2
alloc 4564 4
free 2
free 4
free 1
```

+ Kết quả sau khi chạy lệnh: ./mem input/proc/m0

```
000: 00000-003ff - PID: 01 (idx 000, nxt: 001)
        003e8: 15
001: 00400-007ff - PID: 01 (idx 001, nxt: -01)
002: 00800-00bff - PID: 01 (idx 002, nxt: 003)
003: 00c00-00fff - PID: 01 (idx 003, nxt: 004)
```

004: 01000-013ff - PID: 01 (idx 004, nxt: 005)

005: 01400-017ff - PID: 01 (idx 005, nxt: 006)

006: 01800-01bff - PID: 01 (idx 006, nxt: -01)

014: 03800-03bff - PID: 01 (idx 014, nxt: 015)

03814: 66

015: 03c00-03fff - PID: 01 (idx 015, nxt: -01)

Ý nghĩa kết quả:

Dòng 1: 000: 00000-003ff - PID: 01 (idx 000, nxt: 001)

000 : index của page

003ff: địa chỉ bắt đầu của page

01: ID của process

000: index của page trong danh sách page được cấp phát.

001: index của page kế tiếp trong danh sách của page đươc cấp phát.

Các dòng 3,4,5,6,7,8,9,11 có ý nghĩa tương tự như dòng 1.

Dòng 2: 003e8: 15

003e8: là địa chỉ của vùng nhớ trên RAM.

15: giá trị thuộc vùng nhớ có địa chỉ 003e8 lưu trữ.

Giải thích kết quả:

Trong process m0 có 2 hàng sau:

write 102 1 20

write 21 2 1000

Ý nghĩa dòng thứ nhất :

Ghi giá trị 102 (hệ hex : 66) vào thanh ghi 1 có offset 20 (hệ hex : 14).

Vậy nên dòng 03814: 66 là :

Địa chỉ vùng nhớ trên RAM: 03800 + 14 = 03814

Giá trị là 66.

Dòng thứ 2 tương tự:

Vậy nên dòng 003e8: 15 là:

Địa chỉ vùng nhớ trên RAM : 000 + 3e8 = 003e8

Giá trị là 15.

+ Kết quả sau khi chạy dòng lệnh : ./mem input/proc/m1

Không in ra gì,vì nó đã free những vùng nhớ được cấp phát trong cùng process 1.