

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 (TN) (CO2018)

Bài tập lớn

Simple Operating System

Nhóm 2

Giảng viên hướng dẫn: Trần Trương Tấn Phát
Sinh viên: Huỳnh Nguyên Phúc - 2110451
Phạm Thế Hiếu - 2111213
Cao Minh Quân - 2112109
Phạm Phú Khang - 2110245
Dương Chí Hiếu - 2111172
Email: phuc.huynhdaihocbk94@hcmut.edu.vn

Thành phố Hồ Chí Minh, tháng 5/2023



Mục lục

1	Danh sách thành viên & Phân công công việc	2
2	Mở đầu	3
2.1	Đối tượng nghiên cứu	3
2.2	Mục tiêu nghiên cứu	3
3	Nội dung	3
3.1	Scheduler	3
3.1.1	Câu hỏi	3
3.1.2	Kết quả chạy thử	3
3.2	Memory Management	7
3.2.1	The virtual memory mapping in each process	7
3.2.2	The system physical memory	8
3.2.3	Paging-based address translation scheme	9
3.2.4	Kết quả chạy thử	10
3.2.5	Wrapping up all paging-oriented implementations	12
3.3	Put it all together (Synchronization)	16
3.3.1	Câu hỏi	16
3.3.2	Kết quả chạy thử	16



1 Danh sách thành viên & Phân công công việc

STT	Họ và tên	MSSV	Công việc	Phần trăm công việc
1	Huỳnh Nguyên Phúc	2110451	- Làm phần 2.2.3 (Code + Question) - Làm báo cáo, slide	20%
2	Phạm Thế Hiếu	2111213	- Làm phần 2.2.2 (Code + Question) - Làm báo cáo	18%
3	Cao Minh Quân	2112109	- Làm phần 2.2.1, 2.3 (Code + Question) - Trả lời câu hỏi Synchronization - Làm báo cáo	24%
4	Phạm Phú Khang	2110245	- Làm phần 2.2.4 (Code) - Làm báo cáo	20%
5	Dương Chí Hiếu	2111172	- Làm phần 2.1 (Code) - Trả lời câu hỏi Scheduling - Làm báo cáo	18%

2 Mở đầu

2.1 Đối tượng nghiên cứu

Trong bài tập lớn này, chúng ta sẽ tập trung chủ yếu vào hai phần chính:

- Scheduler: Xác định process nào sẽ được thực hiện trên CPU vào một thời điểm nhất định.
- Virtual Memory engine: Mỗi process sẽ có một vùng trong bộ nhớ ảo riêng và virtual memory engine sẽ thực hiện ánh xạ và biên dịch địa chỉ ảo cung cấp bởi các process tới địa chỉ vật lý tương ứng trên bộ nhớ vật lý.

Bài tập lớn có sử dụng các kiến thức liên quan đến Scheduling, Synchronization và Virtual Memory.

2.2 Mục tiêu nghiên cứu

- Giúp sinh viên hiểu hơn về môn học Hệ điều hành, cụ thể là các nội dung về Scheduling, Synchronization và Virtual Memory
- Giúp sinh viên hiểu được phần nào nguyên lý hoạt động, cách thức hoạt động của một hệ điều hành đơn giản; hiểu được ý nghĩa, vai trò, cách hiện thực của từng module trong hệ điều hành cũng như cách thức hoạt động của nó.

3 Nội dung

3.1 Scheduler

Định thời là một cơ chế của hệ điều hành giúp quản lý việc phân bổ thời gian sử dụng bộ xử lý trung tâm (CPU) cho các tiến trình (processes) hoặc các luồng (threads) đang chạy trên hệ điều hành (Operating System).

3.1.1 Câu hỏi

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

Answer:

Ưu điểm của việc sử dụng priority queue so với các thuật toán định thời khác như FCFS, SJF hay RR là nó cho phép ưu tiên các process dựa trên tầm quan trọng, cấp bách. Điều này cho phép các process quan trọng được ưu tiên hơn các process ít quan trọng hơn, dẫn đến hiệu suất và khả năng đáp ứng của hệ thống tốt hơn.

Thuật toán định thời sử dụng priority queue có thể hiệu quả hơn so với các thuật toán định thời khác vì nó cho phép định thời ưu tiên (preemptive scheduling). Nói cách khác, process ưu tiên cao hơn có thể được lập lịch chạy bất cứ lúc nào, ngay cả khi một process ưu tiên thấp hơn đang chạy. Điều này giúp đảm bảo rằng các process quan trọng được hoàn thành đúng thời hạn, mà không lãng phí tài nguyên cho các tác vụ ưu tiên thấp hơn.

3.1.2 Kết quả chạy thử

Ta chạy thử một số testcase nhằm kiểm tra việc hiện thực giải thuật Multilevel Queue cho Scheduler.



a) File **sched_0**

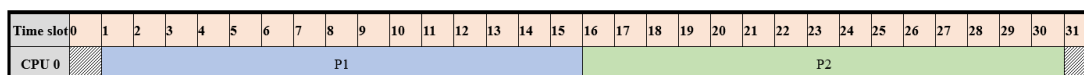
Nội dung file sched_0:

```
input > sched_0
...
1 2 1 2
2 0 s0
3 4 s1
4
```

Kết quả:

```
miniv@DESKTOP-SLLNRQR: ~$ cat sched_0
ld_routine
Loaded a process at input/proc/s0, PID: 1 PRIO: 4
Time slot 1
Loaded a process at input/proc/s0, PID: 2 PRIO: 0
CPU 0: Dispatched process 1
Time slot 2
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 3
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 4
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 5
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 6
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 7
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 8
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 9
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 10
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 11
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 12
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 13
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 14
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 15
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 1
Time slot 16
CPU 0: Processed 1 has finished
CPU 0: Dispatched process 2
Time slot 17
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 18
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 19
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 20
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 21
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 22
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 23
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 24
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 25
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 26
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 27
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 28
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 29
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 30
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 2
Time slot 31
CPU 0: Processed 2 has finished
CPU 0 stopped
```

Gantt Chart:



b) File **sched_1**

Nội dung file sched_1:

```
input > sched_1
...
1 2 1 4
2 0 s0
3 4 s1
4 6 s2
5 7 s3
6
```

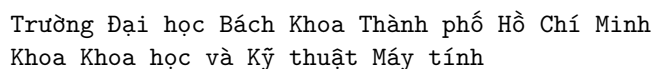
Kết quả:


```
miniv@DESKTOP-SLLNRQR: ~$ x + v miniv@DESKTOP-SLLNRQR: ~$ x + v  
ld_routine  
Time slot 0  
    Loaded a process at input/proc/s0, PID: 1 PRIO: 4  
Time slot 1  
    Loaded a process at input/proc/s0, PID: 2 PRIO: 0  
CPU 0: Dispatched process 2  
    Loaded a process at input/proc/s0, PID: 3 PRIO: 0  
Time slot 2  
CPU 0: Put process 2 to run queue  
CPU 0: Dispatched process 3  
    Loaded a process at input/proc/s0, PID: 4 PRIO: 0  
Time slot 4  
Time slot 5  
CPU 0: Put process 3 to run queue  
CPU 0: Dispatched process 2  
Time slot 6  
Time slot 7  
CPU 0: Put process 2 to run queue  
CPU 0: Dispatched process 4  
Time slot 8  
Time slot 9  
CPU 0: Put process 4 to run queue  
CPU 0: Dispatched process 3  
Time slot 10  
Time slot 11  
CPU 0: Put process 3 to run queue  
CPU 0: Dispatched process 2  
Time slot 12  
Time slot 13  
CPU 0: Put process 2 to run queue  
CPU 0: Dispatched process 4  
Time slot 14  
Time slot 15  
  
Time slot 15  
CPU 0: Put process 4 to run queue  
CPU 0: Dispatched process 3  
Time slot 16  
Time slot 17  
CPU 0: Put process 3 to run queue  
CPU 0: Dispatched process 2  
Time slot 18  
Time slot 19  
CPU 0: Put process 2 to run queue  
CPU 0: Dispatched process 4  
Time slot 20  
Time slot 21  
CPU 0: Put process 4 to run queue  
CPU 0: Dispatched process 3  
Time slot 22  
Time slot 23  
CPU 0: Put process 3 to run queue  
CPU 0: Dispatched process 2  
Time slot 24  
Time slot 25  
CPU 0: Put process 2 to run queue  
CPU 0: Dispatched process 4  
Time slot 26  
Time slot 27  
CPU 0: Put process 4 to run queue  
CPU 0: Dispatched process 3  
Time slot 28  
Time slot 29  
CPU 0: Put process 3 to run queue  
CPU 0: Dispatched process 2  
Time slot 30  
Time slot 31  
CPU 0: Put process 2 to run queue  
  
miniv@DESKTOP-SLLNRQR: ~$ x + v miniv@DESKTOP-SLLNRQR: ~$ x + v  
Time slot 31  
CPU 0: Put process 2 to run queue  
CPU 0: Dispatched process 4  
Time slot 32  
Time slot 33  
CPU 0: Put process 4 to run queue  
CPU 0: Dispatched process 3  
Time slot 34  
Time slot 35  
CPU 0: Put process 3 to run queue  
CPU 0: Dispatched process 2  
Time slot 36  
Time slot 37  
CPU 0: Put process 2 to run queue  
CPU 0: Dispatched process 4  
Time slot 38  
Time slot 39  
CPU 0: Put process 4 to run queue  
CPU 0: Dispatched process 3  
Time slot 40  
Time slot 41  
CPU 0: Put process 3 to run queue  
CPU 0: Dispatched process 2  
Time slot 42  
CPU 0: Processed 2 has finished  
CPU 0: Dispatched process 4  
Time slot 43  
Time slot 44  
CPU 0: Put process 4 to run queue  
CPU 0: Dispatched process 3  
Time slot 45  
CPU 0: Processed 3 has finished  
CPU 0: Dispatched process 4  
Time slot 46  
  
Time slot 46  
CPU 0: Processed 4 has finished  
CPU 0: Dispatched process 1  
Time slot 47  
Time slot 48  
CPU 0: Put process 1 to run queue  
CPU 0: Dispatched process 1  
Time slot 49  
Time slot 50  
CPU 0: Put process 1 to run queue  
CPU 0: Dispatched process 1  
Time slot 51  
Time slot 52  
CPU 0: Put process 1 to run queue  
CPU 0: Dispatched process 1  
Time slot 53  
Time slot 54  
CPU 0: Put process 1 to run queue  
CPU 0: Dispatched process 1  
Time slot 55  
Time slot 56  
CPU 0: Put process 1 to run queue  
CPU 0: Dispatched process 1  
Time slot 57  
Time slot 58  
CPU 0: Put process 1 to run queue  
CPU 0: Dispatched process 1  
Time slot 59  
Time slot 60  
CPU 0: Put process 1 to run queue  
CPU 0: Dispatched process 1  
Time slot 61  
CPU 0: Processed 1 has finished  
CPU 0 stopped
```

Gantt Chart:

[illegible]

c) File `os_1_singleCPU_mfq`
 Nội dung file `os_1_singleCPU_mfq`:



```
input >  os_1_singleCPU_mfq
...
1 2 1 8
2 1 s4 4
3 2 s3 3
4 4 m1s 2
5 6 s2 3
6 7 m0s 3
7 9 p1s 2
8 11 s0 1
9 16 s1 0
10
```

Kết quả:

```
miniv@DESKTOP-SLLNRQR: ~$ x
+
ld_routine
Time slot 0
Time slot 1
    Loaded a process at input/proc/s4, PID: 1 PRIO: 4
    CPU 0: Dispatched process 1
Time slot 2
    Loaded a process at input/proc/s3, PID: 2 PRIO: 3
Time slot 3
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 1
Time slot 4
    Loaded a process at input/proc/mis, PID: 3 PRIO: 2
Time slot 5
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 1
Time slot 6
    Loaded a process at input/proc/s2, PID: 4 PRIO: 3
Time slot 7
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 1
    Loaded a process at input/proc/m0s, PID: 5 PRIO: 3
Time slot 8
    CPU 0: Processed 1 has finished
    CPU 0: Dispatched process 3
    Loaded a process at input/proc/pls, PID: 6 PRIO: 2
Time slot 9
Time slot 10
    CPU 0: Put process 3 to run queue
    CPU 0: Dispatched process 6
    Loaded a process at input/proc/s0, PID: 7 PRIO: 1
Time slot 11
Time slot 12
    CPU 0: Put process 6 to run queue
    CPU 0: Dispatched process 3
miniv@DESKTOP-SLLNRQR: ~$ x
+
Time slot 13
Time slot 14
    CPU 0: Put process 3 to run queue
    CPU 0: Dispatched process 6
Time slot 15
Time slot 16
    CPU 0: Put process 6 to run queue
    CPU 0: Dispatched process 3
    Loaded a process at input/proc/s1, PID: 8 PRIO: 0
Time slot 17
Time slot 18
    CPU 0: Put process 3 to run queue
    CPU 0: Dispatched process 6
Time slot 19
Time slot 20
    CPU 0: Put process 6 to run queue
    CPU 0: Dispatched process 3
Time slot 21
Time slot 22
    CPU 0: Processed 3 has finished
    CPU 0: Dispatched process 6
Time slot 23
Time slot 24
    CPU 0: Put process 6 to run queue
    CPU 0: Dispatched process 6
Time slot 25
Time slot 26
    CPU 0: Processed 6 has finished
    CPU 0: Dispatched process 2
Time slot 27
Time slot 28
    CPU 0: Put process 2 to run queue
    CPU 0: Dispatched process 4
Time slot 29
miniv@DESKTOP-SLLNRQR: ~$ x
+
CPU 0: Put process 4 to run queue
CPU 0: Dispatched process 5
Time slot 31
Time slot 32
    CPU 0: Put process 5 to run queue
    CPU 0: Dispatched process 2
Time slot 33
Time slot 34
    CPU 0: Put process 2 to run queue
    CPU 0: Dispatched process 4
Time slot 35
Time slot 36
    CPU 0: Put process 4 to run queue
    CPU 0: Dispatched process 5
Time slot 37
Time slot 38
    CPU 0: Put process 5 to run queue
    CPU 0: Dispatched process 2
Time slot 39
Time slot 40
    CPU 0: Put process 2 to run queue
    CPU 0: Dispatched process 4
Time slot 41
Time slot 42
    CPU 0: Put process 4 to run queue
    CPU 0: Dispatched process 5
Time slot 43
Time slot 44
    CPU 0: Put process 5 to run queue
    CPU 0: Dispatched process 2
Time slot 45
Time slot 46
    CPU 0: Put process 2 to run queue
    CPU 0: Dispatched process 4
Time slot 47
Time slot 48
    CPU 0: Put process 4 to run queue
    CPU 0: Dispatched process 5
Time slot 49
Time slot 50
    CPU 0: Processed 5 has finished
    CPU 0: Dispatched process 2
Time slot 51
Time slot 52
    CPU 0: Put process 2 to run queue
    CPU 0: Dispatched process 4
Time slot 53
Time slot 54
    CPU 0: Put process 4 to run queue
    CPU 0: Dispatched process 2
Time slot 55
Time slot 56
    CPU 0: Processed 2 has finished
    CPU 0: Dispatched process 4
Time slot 57
Time slot 58
    CPU 0: Processed 4 has finished
    CPU 0: Dispatched process 8
Time slot 59
Time slot 60
    CPU 0: Put process 8 to run queue
    CPU 0: Dispatched process 8
Time slot 61
Time slot 62
    CPU 0: Put process 8 to run queue
    CPU 0: Dispatched process 8
Time slot 63
    CPU 0: Processed 8 has finished
    CPU 0: Dispatched process 7
Time slot 64
Time slot 65
    CPU 0: Put process 7 to run queue
    CPU 0: Dispatched process 7
Time slot 66
Time slot 67
    CPU 0: Put process 7 to run queue
    CPU 0: Dispatched process 7
Time slot 68
Time slot 69
    CPU 0: Put process 7 to run queue
    CPU 0: Dispatched process 7
Time slot 70
Time slot 71
    CPU 0: Put process 7 to run queue
    CPU 0: Dispatched process 7
Time slot 72
Time slot 73
    CPU 0: Put process 7 to run queue
    CPU 0: Dispatched process 7
Time slot 74
Time slot 75
    CPU 0: Put process 7 to run queue
    CPU 0: Dispatched process 7
Time slot 76
Time slot 77
    CPU 0: Put process 7 to run queue
    CPU 0: Dispatched process 7
Time slot 78
    CPU 0: Processed 7 has finished
    CPU 0 stopped
```

Gantt Chart:

Time slot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
CPU 0			P1						P3		P6		P3		P6		P3		P6		P3		P6			P2		P4		
Time slot	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
CPU 0	P5		P2		P4		P5		P2		P4		P5		P2		P4		P5	P2		P4		P2	P4	P8				
Time slot	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78											
CPU 0	P8			P7																										

3.2 Memory Management

Trong cơ chế quản lý bộ nhớ của hệ điều hành, các process sẽ được cấp phát một lượng vùng nhớ nhất định trên bộ nhớ chính để có thể được thực thi bởi CPU. Tùy vào các cơ chế nhất định mà các process sẽ được sử dụng bộ nhớ chính theo các cách khác nhau. Paging là cơ chế chia bộ nhớ vật lý thành các khối bằng nhau gọi là **frame**, cũng như chia không gian địa chỉ luận lý (logical address space) thành các **page**. Hệ điều hành sẽ thực hiện ánh xạ các page và frame với nhau, từ đó cung cấp bộ nhớ cho từng process.

Cơ chế paging được thực hiện dựa trên nhiều thành phần khác nhau. Sau đây là mô tả các thành phần để hiện thực cơ chế paging.

3.2.1 The virtual memory mapping in each process

Trong mỗi process, một bộ nhớ ảo được sử dụng để quản lý địa chỉ cho process đó. Các process sẽ không dùng chung không gian địa chỉ ảo. Không gian này có thể được chia thành nhiều vùng khác nhau gọi là **virtual memory area**. Trong mỗi area lại được chia ra thành các **memory region** nhỏ hơn. Các region có thể không liền kề nhau.

Để có thể thực hiện paging, các memory area được chia thành nhiều trang và có 1 bảng phân trang để quản lý các trang này. Mỗi địa chỉ trong không gian địa chỉ ảo đều tương ứng với một trang bất kỳ, được chia ra làm 2 phần:

- **Page number:** Cho biết chỉ số trang của địa chỉ này, có thể dùng làm chỉ mục cho bảng phân trang để ánh xạ tới frame tương ứng trong bộ nhớ vật lý.
- **Page offset:** Cho biết vị trí của địa chỉ đó trong cùng 1 trang. Khi kết hợp với base address từ bảng phân trang sẽ tạo ra một địa chỉ vật lý.

Question: In this simple OS, we implement a design of multiple memory segments or memory areas in source code declaration. What is the advantage of the proposed design of multiple segments?

Answer:

Mỗi process sẽ có nhiều memory segment hay memory area. Ưu điểm của việc thiết kế nhiều segment cho process:

- Việc chia dữ liệu của process cho nhiều segment giúp cho việc sử dụng bộ nhớ hiệu quả hơn, tránh gom lại thành 1 segment quá lớn.
- Các segment có thể cùng hoặc khác kích thước, làm cho nó linh hoạt trong việc sử dụng bộ nhớ như là cấp phát, thu hồi vùng nhớ...

- Các segment có thể được ánh xạ sang bộ nhớ vật lý với địa chỉ vật lý khác nhau. Điều này khiến cho 1 process không nhất thiết phải lưu trữ trong 1 khối liên tục ở bộ nhớ vật lý.
- Không gặp vấn đề phân mảnh nội.

3.2.2 The system physical memory

Bộ nhớ vật lý được chia ra làm 2 loại: Bộ nhớ RAM và bộ nhớ SWAP. RAM được CPU truy xuất trực tiếp và các process chỉ có thể được thực thi sau khi đã nằm trong RAM. Trong khi đó, SWAP là một thiết bị thứ cấp. Để có thể được chạy bởi CPU, các dữ liệu trong SWAP ứng với process đó phải được nạp lên RAM.

Trong thực tế, kích thước của RAM không đủ lớn để nạp frame cho mọi process. Do đó, cơ chế swapping được sử dụng để chuyển các frame qua lại giữa RAM và bộ nhớ vật lý khác (trong Bài tập lớn này sử dụng SWAP). Điều này giúp cho RAM có thể giải phóng nhiều frame cho các process khác, cũng như cho phép thực thi các chương trình có kích thước lớn hơn nhiều so với kích thước của RAM.

Question: *What will happen if we divide the address to more than 2 levels in the paging memory management system?*

Answer:

Trong hệ thống quản lý bộ nhớ bằng phân trang, khi ta thiết kế địa chỉ thành 2 cấp, ta sẽ có hệ thống phân trang đa cấp. Ở hệ thống này, bộ nhớ ảo sẽ được chia nhỏ thành nhiều vùng địa chỉ riêng biệt ứng với những cấp bậc khác nhau trong bảng phân trang.

Ví dụ: Trong hệ thống 2 cấp, vùng địa chỉ cấp 1 hay còn được gọi là danh mục trang (page directory), sẽ sử dụng để ánh xạ tới những địa chỉ trong vùng địa chỉ cấp 2. Vùng địa chỉ cấp 2 sẽ ánh xạ tới vùng nhớ tương ứng trong bộ nhớ vật lý thông qua bảng phân trang.

Khi một process truy cập bộ nhớ ảo, sẽ truy cập vào vùng địa chỉ cấp trên để tìm ra vùng địa chỉ cấp dưới tương ứng cho đến khi có được địa chỉ trên bộ nhớ vật lý.

Ưu điểm:

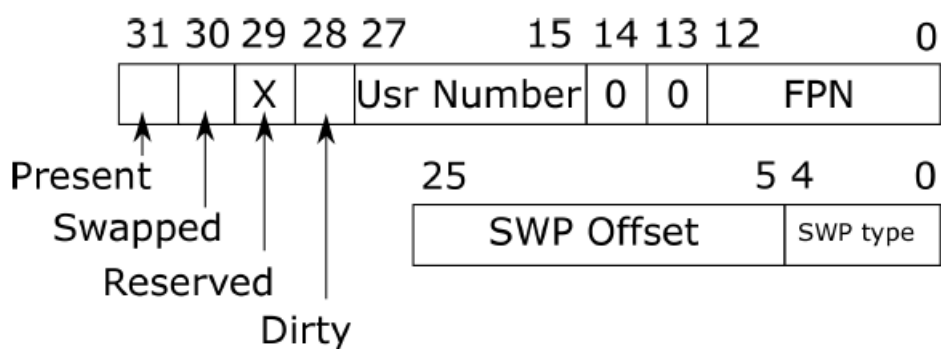
- Quản lý bộ nhớ có địa chỉ lớn.
Ví dụ: Một kiến trúc 32-bit sẽ có 2^{32} byte cần được truy cập tương ứng 2^{32} địa chỉ. Giả sử 1 trang có kích thước là $4KB = 2^{12}$ byte.
Vậy trong thiết kế single-level paging, ta sẽ cần có $2^{32}/2^{12} = 2^{20}$ trang, tương ứng hệ thống phải có một bảng phân trang có 2^{20} entry. Với mỗi entry kích thước 4 byte, ta sẽ có một bảng phân trang có kích thước 4 MB luôn phải dữ trong những process riêng biệt.
Với thiết kế 2-levels, mỗi địa chỉ trong vùng địa chỉ cấp 1 ánh xạ với 1 trang trong vùng địa chỉ cấp 2, vậy với 1 trang trong vùng địa chỉ cấp 1, ta sẽ có $2^{12} \cdot 2^{12} = 2^{24}$ địa chỉ. Vậy để quản lý 2^{32} địa chỉ ta sẽ cần $2^{32}/2^{24} = 2^8$ Trang trong vùng địa chỉ cấp 1. Và mỗi bảng phân trang cấp 2 sẽ có $2^{20}/2^8 = 2^{12}$ entry. Vậy, mỗi process sẽ luôn giữ một bảng phân trang có kích thước $2^8 \cdot 4 = 2^{10} = 1KB$ và load từ vùng nhớ lên một bảng phân trang có kích thước 8 KB. Như vậy là ít hơn rất nhiều so với 4 MB.

Nhược điểm:

- Chi phí thao tác với bộ nhớ là lớn. Vì process chỉ lưu bảng phân trang cấp cao nhất, nên mỗi khi xuống bảng phân trang cấp thấp hơn, process phải load từ bộ nhớ lưu trữ các bảng phân trang đó.

3.2.3 Paging-based address translation scheme

Mỗi process đều có một bảng phân trang để thực hiện ánh xạ các page của nó tới các frame tương ứng trong bộ nhớ vật lý. Cấu trúc của một entry của bảng phân trang trong Bài tập lớn như sau:



Hình 1: Cấu trúc entry của page table

- Bits 0-12 page frame number (FPN) if present
- Bits 13-14 zero if present
- Bits 15-27 user-defined numbering if present
- Bits 0-4 swap type if swapped
- Bits 5-25 swap offset if swapped
- Bit 28 dirty
- Bit 29 reserved
- Bit 30 swapped
- Bit 31 present

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

Answer:

Segmentation with paging là một kỹ thuật quản lý bộ nhớ kết hợp những điểm mạnh của hai kỹ thuật segmentation và paging lại với nhau. Ý tưởng chính của nó sẽ là chia không gian vùng nhớ thành các đơn vị nhỏ hơn, cụ thể là các segments và hơn thế nữa là chia từng segment thành các page khác nhau. Nó giúp việc cấp phát vùng nhớ trở nên linh hoạt hơn. Mỗi segment có thể có nhiều kích thước khác nhau. Nó có những ưu điểm và nhược điểm sau:

Ưu điểm:

- Segmentation with paging giúp cho việc cấp phát vùng nhớ trở nên linh hoạt hơn. Việc phân chia vùng nhớ thành các segments với kích thước tùy ý, cũng như mỗi chương trình hay tiến trình sẽ có tập segments riêng tương ứng sẽ làm cho việc cấp phát vùng nhớ cho các chương trình hay tiến trình sẽ trở nên dễ dàng hơn, dễ đáp ứng hơn các yêu cầu trong việc sử dụng vùng nhớ của chúng, tránh việc xung đột với nhau giữa chúng.

- Segmentation with paging còn giúp chúng ta sử dụng bộ nhớ một cách hiệu quả hơn, giúp cải thiện performance của hệ thống. Nó cho phép chúng ta chỉ cần load những page cần thiết của chương trình, vì thế nên giảm được thời gian thực hiện thao tác swap, cũng như cho phép nhiều chương trình có thể được thực thi đồng thời.
- Giảm hiện tượng phân mảnh ngoại, với việc các segment có thể có kích thước tùy ý.

Nhược điểm:

- Việc hiện thực kỹ thuật segmentation with paging rất phức tạp.
- Việc phải quản lý cùng lúc các segment tables, cũng như các page tables dẫn đến việc tiêu tốn nhiều bộ nhớ và tài nguyên CPU.
- Vẫn còn hiện tượng phân mảnh nội do các page trong từng segment có kích thước cố định.

3.2.4 Kết quả chạy thử

Ta chạy thử testcase **os_1_singleCPU_mlq_paging** nhằm kiểm tra việc hiện thực kỹ thuật Paging cho Memory Management. Testcase này chỉ có 1 CPU nên ta chưa cần quan tâm đến vấn đề đồng bộ.

Nội dung file testcase:

```
input > os_1_singleCPU_mlq_paging
...
1 2 1 8
2 1048576 16777216 0 0 0
3 1 s4 4
4 2 s3 3
5 4 m1s 2
6 6 s2 3
7 7 m0s 3
8 9 p1s 2
9 11 s0 1
10 16 s1 0
11
```

Kết quả:



```
Time slot 0
ld_routine
Time slot 1
  Loaded a process at input/proc/s4, PID: 1 PRIO: 4
  CPU 0: Dispatched process 1
Time slot 2
  Loaded a process at input/proc/s3, PID: 2 PRIO: 3
Time slot 3
  CPU 0: Put process 1 to run queue
  CPU 0: Dispatched process 1
Time slot 4
  Loaded a process at input/proc/mis, PID: 3 PRIO: 2
Time slot 5
  CPU 0: Put process 1 to run queue
  CPU 0: Dispatched process 1
Time slot 6
  Loaded a process at input/proc/s2, PID: 4 PRIO: 3
Time slot 7
  CPU 0: Put process 1 to run queue
  CPU 0: Dispatched process 1
  Loaded a process at input/proc/m0s, PID: 5 PRIO: 3
Time slot 8
  CPU 0: Processed 1 has finished
  CPU 0: Dispatched process 3
Time slot 9
  Loaded a process at input/proc/p1s, PID: 6 PRIO: 2
Time slot 10
  CPU 0: Put process 3 to run queue
  CPU 0: Dispatched process 6
Time slot 11
  Loaded a process at input/proc/s0, PID: 7 PRIO: 1
Time slot 12
  CPU 0: Put process 6 to run queue
  CPU 0: Dispatched process 3
Time slot 13
Time slot 14
  CPU 0: Put process 3 to run queue
  CPU 0: Dispatched process 6
Time slot 15
Time slot 16
  CPU 0: Put process 6 to run queue
  CPU 0: Dispatched process 3
  Loaded a process at input/proc/s1, PID: 8 PRIO: 0
Time slot 17
  CPU 0: Put process 3 to run queue
  CPU 0: Dispatched process 6
Time slot 18
Time slot 19
  CPU 0: Put process 6 to run queue
  CPU 0: Dispatched process 3
Time slot 20
Time slot 21
Time slot 22
  CPU 0: Processed 3 has finished
  CPU 0: Dispatched process 6
Time slot 23
Time slot 24
  CPU 0: Put process 6 to run queue
  CPU 0: Dispatched process 6
Time slot 25
Time slot 26
  CPU 0: Processed 6 has finished
  CPU 0: Dispatched process 2
Time slot 27
Time slot 28
  CPU 0: Put process 2 to run queue
  CPU 0: Dispatched process 4
Time slot 29
Time slot 30
  CPU 0: Put process 4 to run queue
  CPU 0: Dispatched process 5
Time slot 31
Time slot 32
  CPU 0: Put process 5 to run queue
  CPU 0: Dispatched process 2
Time slot 33
```

```
Time slot 34
  CPU 0: Put process 2 to run queue
  CPU 0: Dispatched process 4
Time slot 35
Time slot 36
  CPU 0: Put process 4 to run queue
  CPU 0: Dispatched process 5
Time slot 37
Time slot 38
  CPU 0: Put process 5 to run queue
  CPU 0: Dispatched process 2
Time slot 39
Time slot 40
  CPU 0: Put process 2 to run queue
  CPU 0: Dispatched process 4
Time slot 41
Time slot 42
  CPU 0: Put process 4 to run queue
  CPU 0: Dispatched process 5
write region=1 offset=20 value=102
print_pgtbl: 0 - 512
00000000: 80000003
00000004: c0000000
Memory Dump-----
-----End dump
Time slot 43
write region=2 offset=1000 value=1
print_pgtbl: 0 - 512
00000000: 80000003
00000004: 80000002
Memory Dump-----
Index 64: 102
-----End dump
Time slot 44
  CPU 0: Put process 5 to run queue
  CPU 0: Dispatched process 2
Time slot 45
Time slot 46
  CPU 0: Put process 2 to run queue
  CPU 0: Dispatched process 4
Time slot 47
Time slot 48
  CPU 0: Put process 4 to run queue
  CPU 0: Dispatched process 5
write region=0 offset=0 value=0
print_pgtbl: 0 - 512
00000000: 80000003
00000004: c0000000
Memory Dump-----
Index 64: 102
Index 232: 1
-----End dump
Time slot 49
  CPU 0: Processed 5 has finished
  CPU 0: Dispatched process 2
Time slot 50
Time slot 51
  CPU 0: Put process 2 to run queue
  CPU 0: Dispatched process 4
Time slot 52
Time slot 53
  CPU 0: Put process 4 to run queue
  CPU 0: Dispatched process 2
Time slot 54
  CPU 0: Processed 2 has finished
  CPU 0: Dispatched process 4
Time slot 55
Time slot 56
  CPU 0: Processed 4 has finished
  CPU 0: Dispatched process 8
Time slot 57
Time slot 58
  CPU 0: Put process 8 to run queue
  CPU 0: Dispatched process 8
Time slot 59
Time slot 60
  CPU 0: Put process 8 to run queue
  CPU 0: Dispatched process 8
Time slot 61
Time slot 62
  CPU 0: Put process 8 to run queue
  CPU 0: Dispatched process 8
Time slot 63
  CPU 0: Processed 8 has finished
  CPU 0: Dispatched process 7
Time slot 64
Time slot 65
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 7
Time slot 66
Time slot 67
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 7
Time slot 68
Time slot 69
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 7
Time slot 70
Time slot 71
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 7
Time slot 72
Time slot 73
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 7
Time slot 74
Time slot 75
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 7
Time slot 76
Time slot 77
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 7
Time slot 78
  CPU 0: Processed 7 has finished
  CPU 0 stopped
```

Gantt Chart:



Time slot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CPU		s4 PID=1							m1s PID=3	p1s PID=6	m1s PID=3	p1s PID=6	m1s PID=3	p1s PID=6	m1s PID=3	p1s PID=6	m1s PID=3	p1s PID=6	m1s PID=3	p1s PID=6	m1s PID=3
Time slot	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
CPU	m1s PID=3	p1s PID=6				s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2
Time slot	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
CPU	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	m0s PID=5	s3 PID=2	s2 PID=4	s1 PID=8					
Time slot	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78					
CPU	s0 PID=7																				

Hình 2: Sơ đồ Gantt cho testcase os_1_singleCPU_mlq_paging

3.2.5 Wrapping up all paging-oriented implementations

Trong bài tập lớn này, hệ thống sẽ sử dụng thuật toán MultiLevel Queue cho phần Scheduler và kỹ thuật Memory Paging cho phần Memory Management. Tuy nhiên chúng ta vẫn có thể hoàn toàn không sử dụng các "chế độ mới" đó, sử dụng các "chế độ cũ" mà không cần phải thay đổi sâu vào source code, tránh gây ra đụng độ và vấn đề về đụng độ giữa các phần với nhau. Để làm được điều này, ta sẽ áp dụng cơ chế điều khiển cấu hình hệ thống thông qua các hằng số được định nghĩa từ trước (the configuration control using constant definition).

Các hằng số được định nghĩa trong file *include/os-cfg.h*:

- **MLQ_SCHED** : Sử dụng thuật toán MultiLevel Queue cho phần Scheduler; nếu không được bật thì hệ thống sẽ sử dụng thuật toán Scheduler của phiên bản cũ và đọc file input với định dạng khác.

```
[time slice] [N = Number of CPU] [M = Number of Processes to be run]
[time 0] [path 0]
[time 1] [path 1]
...
[time M-1] [path M-1]
```

- **MM_PAGING** : Sử dụng kỹ thuật Memory Paging cho phần Memory Management; nếu không được bật thì hệ thống sẽ sử dụng kỹ thuật của phiên bản cũ.
- **MM_FIXED_MEMSZ** : Thiết lập cho kích thước của các bộ nhớ là hằng số cho trước; nếu không được bật thì kích thước của các bộ nhớ sẽ được đọc từ file input.
- Ngoài ra còn các hằng số khác dùng để debug, kiểm tra các giá trị:
IODUMP, PAGETBL_DUMP,...

Nhóm sẽ tiến hành chạy thử hệ thống với cấu hình sử dụng phiên bản cũ của hệ thống.

1. Kiểm tra cấu hình **MLQ_SCHED**:

- Cấu hình:

```

1  ✓ #ifndef OSCFG_H
2    #define OSCFG_H
3
4    // #define MLQ_SCHED 1
5    #define MAX_PRIO 140
6
7  ✓ // #define MM_PAGING
8    // #define MM_FIXED_MEMSZ
9    // #define VMDBG 1
10   // #define MMDBG 1
11   #define IODUMP 1
12   #define PAGETBL_DUMP 1
13
14   #endif

```

- Testcase:

```

1    4 2 3
2    0 s1
3    1 s2
4    2 s3

```

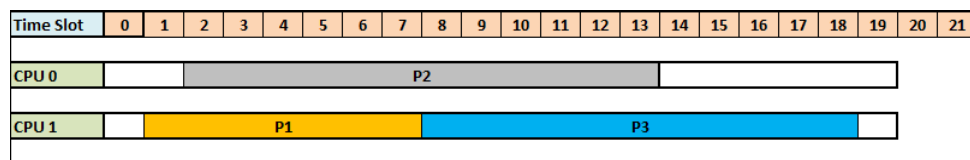
- Kết quả:

```

Time slot 0      ld_routine
                  Loaded a process at input/proc/s1, PID: 1 PRIO: 20
Time slot 1      CPU 1: Dispatched process 1
                  Loaded a process at input/proc/s2, PID: 2 PRIO: 20
Time slot 2      CPU 0: Dispatched process 2
                  Loaded a process at input/proc/s3, PID: 3 PRIO: 7
Time slot 3
Time slot 4
Time slot 5      CPU 1: Put process 1 to run queue
                  CPU 1: Dispatched process 1
Time slot 6      CPU 0: Put process 2 to run queue
                  CPU 0: Dispatched process 2
Time slot 7
Time slot 8      CPU 1: Processed 1 has finished
                  CPU 1: Dispatched process 3
Time slot 9
Time slot 10     CPU 0: Put process 2 to run queue
                  CPU 0: Dispatched process 2
Time slot 11
Time slot 12     CPU 1: Put process 3 to run queue
                  CPU 1: Dispatched process 3
Time slot 13
Time slot 14     CPU 0: Processed 2 has finished
                  CPU 0 stopped
Time slot 15
Time slot 16     CPU 1: Put process 3 to run queue
                  CPU 1: Dispatched process 3
Time slot 17
Time slot 18
Time slot 19     CPU 1: Processed 3 has finished
                  CPU 1 stopped

```

- Gantt Chart:



2. Kiểm tra cấu hình *MM_PAGING*:

- Cấu hình:

```
1  ✓ #ifndef OSCFG_H
2  #define OSCFG_H
3
4  #define MLQ_SCHED 1
5  #define MAX_PRIO 140
6
7  ✓ // #define MM_PAGING
8  // #define MM_FIXED_MEMSZ
9  // #define VMDBG 1
10 // #define MMDBG 1
11 #define IODUMP 1
12 #define PAGETBL_DUMP 1
13
14 #endif
```

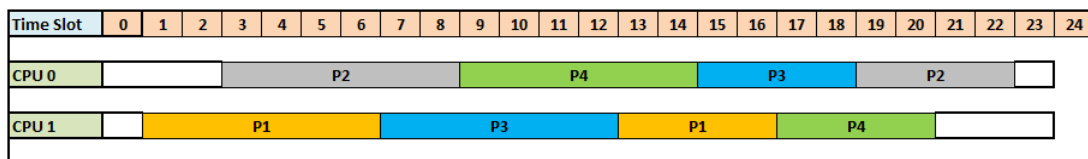
- Testcase:

```
1  6 2 4
2  0 p0s 0
3  2 p1s 15
```

- Kết quả:

```
Time slot 0      ld_routine
                  Loaded a process at input/proc/p0s, PID: 1 PRI0: 0
Time slot 1      CPU 1: Dispatched process 1
Time slot 2      Loaded a process at input/proc/pls, PID: 2 PRI0: 15
Time slot 3      CPU 0: Dispatched process 2
                  Loaded a process at input/proc/pls, PID: 3 PRI0: 0
Time slot 4      Loaded a process at input/proc/pls, PID: 4 PRI0: 0
Time slot 5
Time slot 6
Time slot 7      CPU 1: Put process 1 to run queue
                  CPU 1: Dispatched process 3
Time slot 8
Time slot 9      CPU 0: Put process 2 to run queue
                  CPU 0: Dispatched process 4
Time slot 10
Time slot 11
Time slot 12      Time slot 12
Time slot 13      CPU 1: Put process 3 to run queue
                  CPU 1: Dispatched process 1
Time slot 14
Time slot 15      CPU 0: Put process 4 to run queue
                  CPU 0: Dispatched process 3
Time slot 16
Time slot 17      CPU 1: Processed 1 has finished
                  CPU 1: Dispatched process 4
Time slot 18
Time slot 19      CPU 0: Processed 3 has finished
                  CPU 0: Dispatched process 2
Time slot 20
Time slot 21      CPU 1: Processed 4 has finished
                  CPU 1 stopped
Time slot 22
Time slot 23      CPU 0: Processed 2 has finished
                  CPU 0 stopped
```

- Gantt Chart:



3. Kiểm tra cấu hình *MM_FIXED_MEMSZ*:

- Cấu hình:

```
1  #ifndef OSCFG_H
2  #define OSCFG_H
3
4  #define MLQ_SCHED 1
5  #define MAX_PRIO 140
6
7  #define MM_PAGING
8  #define MM_FIXED_MEMSZ
9  // #define VMDBG 1
10 // #define MMDBG 1
11 #define IODUMP 1
12 #define PAGETBL_DUMP 1
13
14 #endif
```

- Testcase:

```
1  6 2 4
2  0 p0s 0
3  2 p1s 15
```

- Kết quả:

```
Time slot 0
ld_routine
Loaded a process at input/proc/p0s, PID: 1 PRIO: 0
Time slot 1
CPU 0: Dispatched process 1
Time slot 2
Loaded a process at input/proc/p1s, PID: 2 PRIO: 15
Time slot 3
CPU 1: Dispatched process 2
Loaded a process at input/proc/p1s, PID: 3 PRIO: 0
Time slot 4
Loaded a process at input/proc/p1s, PID: 4 PRIO: 0
Time slot 5
Time slot 6
write region=1 offset=20 value=100
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
-----End dump
Time slot 7
CPU 0: Put process 1 to run queue
CPU 0: Dispatched process 3
Time slot 8
Time slot 9
CPU 1: Put process 2 to run queue
CPU 1: Dispatched process 4
Time slot 10
Time slot 11
Time slot 12

Time slot 13
CPU 0: Put process 3 to run queue
CPU 0: Dispatched process 1
read region=1 offset=20 value=100
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 100
-----End dump
Time slot 14
write region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 100
-----End dump
Time slot 15
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 3
read region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 102
-----End dump
Time slot 16
write region=3 offset=20 value=103
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 102
-----End dump
CPU 0: Processed 1 has finished
CPU 0: Dispatched process 4
Time slot 17
Time slot 18
Time slot 19
CPU 1: Processed 3 has finished
CPU 1: Dispatched process 2
Time slot 20
Time slot 21
CPU 0: Processed 4 has finished
CPU 0 stopped
Time slot 22
Time slot 23
CPU 1: Processed 2 has finished
CPU 1 stopped
```

- Gantt Chart:

Time Slot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
CPU 0																									
CPU 1																									

3.3 Put it all together (Synchronization)

3.3.1 Câu hỏi

Trong Bài tập lớn, hệ điều hành được xây dựng để được chạy trên nhiều CPU khác nhau. Các CPU có thể được chạy đồng thời với nhau, và nó đều sử dụng chung 1 bộ nhớ RAM, các bộ nhớ SWAP, hay các queue chứa process control block... Các tài nguyên cần được bảo vệ nhằm tránh xảy ra tranh chấp.

Question: What will happen if the synchronization is not handled in your simple OS? Illustrate by example the problem of your simple OS if you have any.

Answer: Trong hệ điều hành của Bài tập lớn, nếu không xử lý vấn đề đồng bộ, các tài nguyên dùng chung bị thao tác bởi nhiều luồng CPU sẽ gặp phải hiện tượng race condition. Khi đó, dữ liệu bên trong các tài nguyên này có thể bị mất tính toàn vẹn, nhất quán, logic chương trình thực thi sẽ không còn đúng. Ví dụ: Nếu ta không kiểm soát đồng bộ cho ready_queue, các CPU sẽ truy cập đồng thời vào nó và có thể lấy ra cùng lúc 1 pcb, hoặc là 1 CPU đang thêm pcb vào thì lại có 1 CPU khác can thiệp, khiến cho quá trình này không được hoàn chỉnh. Tương tự, nếu không kiểm soát tốt đồng bộ cho memory, các CPU có thể cùng lúc truy cập vào 1 vị trí trên memory ảnh hưởng đến việc quản lý bộ nhớ.

Để hiện thực cơ chế đồng bộ cho hệ điều hành, nhóm đã sử dụng Mutex lock của thư viện Pthreads. Một mutex được sử dụng để bảo vệ ready_queue (**queue_lock**), và một mutex được sử dụng để bảo vệ RAM và SWAP (**mem_lock**). Trước khi thực hiện các thao tác lấy hay thêm pcb vào queue, câu lệnh `pthread_mutex_lock(&queue_lock)` được thực thi nhằm không cho các CPU khác truy cập vào ready_queue. Sau khi thao tác xong với ready_queue, CPU đó sẽ giải phóng mutex bằng `pthread_mutex_unlock(&queue_lock)`.

Tương tự, trước khi thao tác với RAM và SWAP trong các lệnh alloc/free/read/write, câu lệnh `pthread_mutex_lock(&mem_lock)` được thực thi nhằm ngăn các CPU khác truy cập đồng thời vào RAM. Sau khi thực hiện xong với bộ nhớ, CPU này sẽ giải phóng mutex bằng `pthread_mutex_unlock(&mem_lock)`.

3.3.2 Kết quả chạy thử

Sau khi hoàn thành xong các thành phần của hệ điều hành (gồm có Scheduler, Memory Management và Synchronization), nhóm sẽ chạy thử một số testcase nhằm kiểm tra toàn bộ hệ điều hành.

a) File `os_0_mmq_paging`

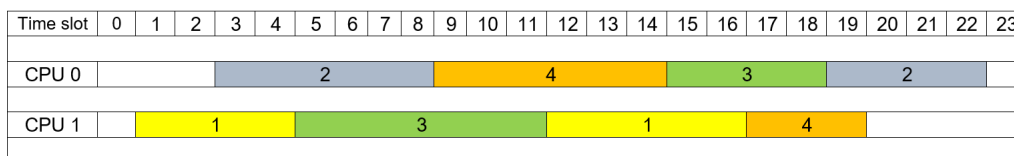
Nội dung file testcase:

```
input > os_0_mlq_paging
...
1 6 2 4
2 1048576 16777216 0 0 0
3 0 p0s 0
4 2 p1s 15
```

Kết quả:

```
minv@DESKTOP-5LLNRQR: ~$ cat
Time slot 0
ld_routine
Loaded a process at input/proc/p0s, PID: 1 PRI0: 0
Time slot 1
CPU 1: Dispatched process 1
Time slot 2
Loaded a process at input/proc/p1s, PID: 2 PRI0: 15
Time slot 3
CPU 0: Dispatched process 2
Loaded a process at input/proc/p1s, PID: 3 PRI0: 0
Time slot 4
Loaded a process at input/proc/p1s, PID: 4 PRI0: 0
Time slot 5
write region=1 offset=20 value=100
print_pgtbl: 0 - 1024Time slot 6
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
-----End dump
CPU 1: Put process 1 to run queue
CPU 1: Dispatched process 3
Time slot 7
Time slot 8
Time slot 9
CPU 0: Put process 2 to run queue
CPU 0: Dispatched process 4
Time slot 10
Time slot 11
Time slot 12
CPU 1: Put process 3 to run queue
CPU 1: Dispatched process 1
Time slot 13
read region=1 offset=20 value=100
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 100
-----End dump
Time slot 14
write region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 100
-----End dump
Time slot 15
CPU 0: Put process 4 to run queue
CPU 0: Dispatched process 3
read region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 102
-----End dump
Time slot 16
write region=3 offset=20 value=103
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 102
-----End dump
Time slot 17
CPU 1: Processed 1 has finished
CPU 1: Dispatched process 4
Time slot 18
Time slot 19
CPU 0: Processed 3 has finished
CPU 0: Dispatched process 2
Time slot 20
CPU 1: Processed 4 has finished
CPU 1 stopped
Time slot 21
Time slot 22
Time slot 23
CPU 0: Processed 2 has finished
CPU 0 stopped
```

Gantt Chart:



b) File `os_1_mlq_paging`
Nội dung file testcase:



```
input > os_1_mfq_paging
...
1 2 4 8
2 1048576 16777216 0 0 0
3 1 p0s 130
4 2 s3 39
5 4 m1s 15
6 6 s2 120
7 7 m0s 120
8 9 p1s 15
9 11 s0 38
10 16 s1 0
11
```

Kết quả:

```
minv@DESKTOP-SLLNRQ: ~. x + v minv@DESKTOP-SLLNRQ: ~. x + v
Time slot 0 ld_routine
Time slot 1 Loaded a process at input/proc/p0s, PID: 1 PRIO: 130
CPU 3: Dispatched process 1
Time slot 2 Loaded a process at input/proc/s3, PID: 2 PRIO: 39
Time slot 3 CPU 1: Dispatched process 2
CPU 3: Put process 1 to run queue
CPU 3: Dispatched process 1
Time slot 4 Loaded a process at input/proc/m1s, PID: 3 PRIO: 15
CPU 2: Dispatched process 3
CPU 1: Put process 2 to run queue
Time slot 5 CPU 1: Dispatched process 2
CPU 3: Put process 1 to run queue
CPU 3: Dispatched process 1
Time slot 6 Loaded a process at input/proc/s2, PID: 4 PRIO: 120
write region=1 offset=20 value=100
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
CPU 2: Put process 3 to run queue
CPU 0: Dispatched process 4
CPU 2: Dispatched process 3
Time slot 7 Memory Dump-----
CPU 1: Put process 2 to run queue
CPU 1: Dispatched process 2
Time slot 7 Time slot 7
Memory Dump-----
CPU 1: Put process 2 to run queue
CPU 1: Dispatched process 2
Loaded a process at input/proc/m0s, PID: 5 PRIO: 120
-----End dump
CPU 3: Put process 1 to run queue
CPU 3: Dispatched process 5
Time slot 8 CPU 2: Put process 3 to run queue
CPU 0: Put process 4 to run queue
CPU 0: Dispatched process 4
CPU 2: Dispatched process 1
read region=1 offset=20 value=100
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 100
CPU 1: Put process 2 to run queue
CPU 1: Dispatched process 3
Time slot 9 Loaded a process at input/proc/p1s, PID: 6 PRIO: 15
-----End dump
write region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 100
```



```
minv@DESKTOP-SLLNRQR: ~. x + v minv@DESKTOP-SLLNRQR: ~. x + v
Time slot 10 CPU 3: Put process 5 to run queue
CPU 3: Dispatched process 6
-----End dump
Time slot 11 CPU 2: Put process 1 to run queue
CPU 2: Dispatched process 2
CPU 0: Put process 4 to run queue
CPU 0: Dispatched process 5
CPU 1: Put process 3 to run queue
CPU 1: Dispatched process 4
Loaded a process at input/proc/s0, PID: 7 PRIO: 38
CPU 3: Put process 6 to run queue
Time slot 12 CPU 3: Dispatched process 1
read region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 102
-----End dump
CPU 2: Put process 2 to run queue
CPU 2: Dispatched process 3
write region=3 offset=20 value=103
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 102
Index 20: 102
CPU 0: Put process 5 to run queue
CPU 0: Dispatched process 6
CPU 0: Put process 4 to run queue
CPU 0: Dispatched process 6
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 7
Time slot 13 CPU 0: Put process 5 to run queue
CPU 0: Dispatched process 6
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 7
Time slot 14 CPU 2: Processed 1 has finished
CPU 3: Dispatched process 2
Time slot 15 CPU 2: Processed 3 has finished
CPU 2: Dispatched process 5
write region=1 offset=20 value=102
print_pgtbl: 0 - 512 CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 4
00000000: 80000007
00000004: 80000006
Memory Dump-----
Index 20: 103
Time slot 16 CPU 0: Put process 6 to run queue
CPU 0: Dispatched process 6
-----End dump
CPU 3: Put process 2 to run queue
CPU 3: Dispatched process 7
write region=2 offset=1000 value=1
print_pgtbl: 0 - 512
00000000: 80000007
00000004: 80000006
Memory Dump-----
Index 20: 103
Index 64: 102
Time slot 17 Loaded a process at input/proc/s1, PID: 8 PRIO: 0
-----End dump
CPU 2: Put process 5 to run queue
CPU 2: Dispatched process 2
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 5
Time slot 18 CPU 0: Put process 6 to run queue
CPU 0: Dispatched process 4
-----End dump
CPU 2: Processed 2 has finished
CPU 3: Put process 7 to run queue
CPU 3: Dispatched process 6
CPU 1: Processed 5 has finished
CPU 1: Dispatched process 7
CPU 2: Dispatched process 8
Time slot 19 CPU 0: Put process 4 to run queue
CPU 0: Dispatched process 4
Time slot 20 CPU 3: Put process 6 to run queue
CPU 3: Dispatched process 6
CPU 2: Put process 8 to run queue
CPU 2: Dispatched process 8
CPU 3: Dispatched process 6
CPU 2: Put process 8 to run queue
CPU 2: Dispatched process 8
CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
Time slot 21 CPU 0: Processed 4 has finished
CPU 0 stopped
CPU 3: Processed 6 has finished
CPU 3 stopped
Time slot 22 CPU 2: Put process 8 to run queue
CPU 2: Dispatched process 8
CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
Time slot 23 CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
CPU 2: Put process 8 to run queue
CPU 2: Dispatched process 8
CPU 2: Processed 8 has finished
CPU 2 stopped
Time slot 24 CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
Time slot 25 CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
Time slot 26 CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
Time slot 27 CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
Time slot 28 CPU 1: Put process 7 to run queue
CPU 1: Dispatched process 7
Time slot 29 CPU 1: Processed 7 has finished
CPU 1 stopped
```

Gantt Chart:



Time slot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
CPU 0									4		5				6			4						
CPU 1			2			2		3		4		7		4		5			7					
CPU 2					3			1		2		3		5		2			8					
CPU 3				1				5		6			1		2		7		6					

Time slot	24	25	26	27	28	29
CPU 0						
CPU 1			7			
CPU 2						
CPU 3						

c) File `os_1_mlq_paging_small_1K`

Nội dung file testcase:

```
input > os_1_mlq_paging_small_1K
...
1 2 4 8
2 2048 16777216 0 0 0
3 1 p0s 130
4 2 s3 39
5 4 m1s 15
6 6 s2 120
7 7 m0s 120
8 9 p1s 15
9 11 s0 38
10 16 s1 0
11
```

Kết quả:



```
Time slot 0
ld_routine
Time slot 1
  Loaded a process at input/proc/p0s, PID: 1 PRIO: 130
  CPU 3: Dispatched process 1
Time slot 2
  Loaded a process at input/proc/s3, PID: 2 PRIO: 39
Time slot 3
  CPU 1: Dispatched process 2
  CPU 3: Put process 1 to run queue
  CPU 3: Dispatched process 1
Time slot 4
  Loaded a process at input/proc/mis, PID: 3 PRIO: 15
Time slot 5
  CPU 1: Put process 2 to run queue
  CPU 1: Dispatched process 3
  CPU 0: Dispatched process 2
  CPU 3: Put process 1 to run queue
  CPU 3: Dispatched process 1
Time slot 6
  Loaded a process at input/proc/s2, PID: 4 PRIO: 120
  write region=1 offset=20 value=100
  print_pgtbl: 0 - 1024
  00000000: 80000001
  00000004: 80000000
  00000008: 80000003
  00000012: 80000002
  Memory Dump-----
  -----End dump
Time slot 7
  CPU 1: Put process 3 to run queue
  CPU 1: Dispatched process 3
  CPU 0: Put process 2 to run queue
  CPU 0: Dispatched process 2
  CPU 2: Dispatched process 4
  Loaded a process at input/proc/m0s, PID: 5 PRIO: 120
  CPU 3: Put process 1 to run queue
  CPU 3: Dispatched process 5
Time slot 8
Time slot 9
  CPU 1: Put process 3 to run queue
  CPU 1: Dispatched process 1
  read region=1 offset=20 value=100
Time slot 9
  CPU 1: Put process 3 to run queue
  CPU 1: Dispatched process 1
  read region=1 offset=20 value=100
  print_pgtbl: 0 - 1024
  00000000: 80000001
  00000004: 80000000
  00000008: 80000003
  00000012: 80000002
  Memory Dump-----
  Index 20: 100
  -----End dump
Time slot 10
  write region=2 offset=20 value=102
  print_pgtbl: 0 - 1024
  00000000: 80000001
  00000004: 80000000
  00000008: 80000003
  00000012: 80000002
  Memory Dump-----
  Index 20: 100
  -----End dump
Time slot 11
  CPU 1: Put process 1 to run queue
  CPU 1: Dispatched process 5
  CPU 0: Put process 3 to run queue
  CPU 0: Dispatched process 1
  read region=2 offset=20 value=102
  print_pgtbl: 0 - 1024
  00000000: 80000001
  00000004: 80000000
  00000008: 80000003
  00000012: 80000002
  Memory Dump-----
  Index 20: 102
  -----End dump
```

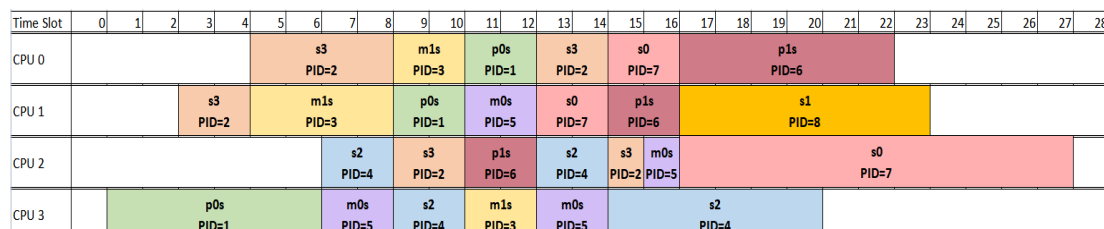
```
Time slot 12
  write region=3 offset=20 value=103
  print_pgtbl: 0 - 1024
  00000000: 80000001
  00000004: 80000000
  00000008: 80000003
  00000012: 80000002
  Memory Dump-----
  Index 20: 102
  -----End dump
Time slot 13
  CPU 1: Put process 5 to run queue
  CPU 1: Dispatched process 7
  CPU 0: Processed 1 has finished
  CPU 0: Dispatched process 2
  CPU 2: Put process 6 to run queue
  CPU 2: Dispatched process 4
  CPU 3: Processed 3 has finished
  CPU 3: Dispatched process 5
  write region=1 offset=20 value=102
  print_pgtbl: 0 - 512
  00000000: 80000007
  00000004: 80000006
  Memory Dump-----
  Index 20: 103
  Index 64: 102
  -----End dump
Time slot 14
  write region=2 offset=1000 value=1
  print_pgtbl: 0 - 512
  00000000: 80000007
  00000004: 80000006
  Memory Dump-----
  Index 20: 103
  Index 64: 102
  -----End dump
Time slot 15
Time slot 15
  CPU 1: Put process 7 to run queue
  CPU 1: Dispatched process 6
  CPU 0: Put process 2 to run queue
  CPU 0: Dispatched process 7
  CPU 2: Put process 4 to run queue
  CPU 2: Dispatched process 2
  CPU 3: Put process 5 to run queue
  CPU 3: Dispatched process 4
Time slot 16
  CPU 2: Processed 2 has finished
  CPU 2: Dispatched process 5
  write region=0 offset=0 value=0
  print_pgtbl: 0 - 512
  00000000: 80000007
  00000004: c0000000
  Memory Dump-----
  Index 20: 103
  Index 64: 102
  Index 232: 1
  -----End dump
  Loaded a process at input/proc/s1, PID: 8 PRIO: 0
Time slot 17
  CPU 1: Put process 6 to run queue
  CPU 1: Dispatched process 8
  CPU 0: Put process 7 to run queue
  CPU 0: Dispatched process 6
  CPU 2: Processed 5 has finished
  CPU 2: Dispatched process 7
  CPU 3: Put process 4 to run queue
  CPU 3: Dispatched process 4
Time slot 18
Time slot 19
  CPU 1: Put process 8 to run queue
  CPU 1: Dispatched process 8
  CPU 0: Put process 6 to run queue
  CPU 0: Dispatched process 6
  CPU 2: Put process 7 to run queue
  CPU 2: Dispatched process 7
  CPU 3: Put process 4 to run queue
  CPU 3: Dispatched process 4
```

```

Time slot 20
Time slot 21
    CPU 1: Put process 8 to run queue
    CPU 1: Dispatched process 8
    CPU 0: Put process 6 to run queue
    CPU 0: Dispatched process 6
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
    CPU 3: Processed 4 has finished
    CPU 3 stopped
Time slot 22
Time slot 23
    CPU 1: Put process 8 to run queue
    CPU 1: Dispatched process 8
    CPU 0: Processed 6 has finished
    CPU 0 stopped
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 24
    CPU 1: Processed 8 has finished
    CPU 1 stopped
Time slot 25
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 26
Time slot 27
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 28
    CPU 2: Processed 7 has finished
    CPU 2 stopped

```

Gantt Chart:



Hình 3: Sơ đồ Gantt cho testcase os_1_mlq_paging_small_1K

d) File **os_1_mlq_paging_small_4K**

Nội dung file testcase:



```
input > os_1_mfq_paging_small_4K
...
1 2 4 8
2 4096 16777216 0 0 0
3 1 p0s 130
4 2 s3 39
5 4 m1s 15
6 6 s2 120
7 7 m0s 120
8 9 p1s 15
9 11 s0 38
10 16 s1 0
11
```

Kết quả:

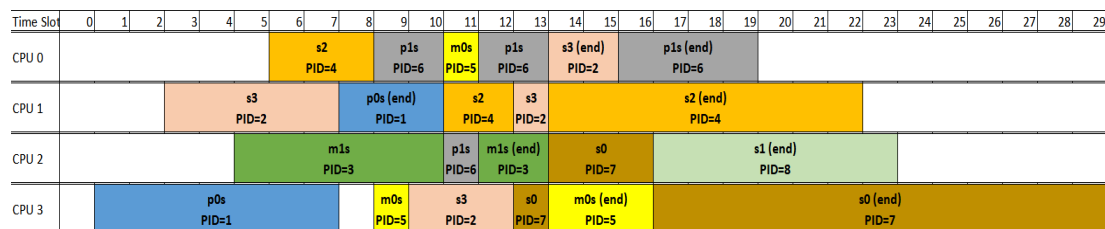
```
minv@DESKTOP-5LLNRQR: ~. X + v minv@DESKTOP-5LLNRQR: ~. X + v
Time slot 0 CPU 2: Dispatched process 3
ld_routine CPU 3: Put process 1 to run queue
Time slot 1 Loaded a process at input/proc/p0s, PID: 1 PRI0: 130 CPU 0: Dispatched process 4
CPU 3: Dispatched process 1 Loaded a process at input/proc/m0s, PID: 5 PRI0: 120
CPU 3: Dispatched process 1
Time slot 2 read region=1 offset=20 value=100
Loaded a process at input/proc/s3, PID: 2 PRI0: 39 print_pgtbl: 0 - 1024
CPU 1: Dispatched process 2 00000000: 80000001
CPU 3: Put process 1 to run queue Time slot 7
CPU 3: Dispatched process 1 00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 4 Memory Dump-----
Loaded a process at input/proc/m1s, PID: 3 PRI0: 15 Index 20: 100
CPU 1: Put process 2 to run queue CPU 1: Put process 2 to run queue
CPU 1: Dispatched process 2 CPU 1: Dispatched process 2
CPU 2: Dispatched process 3 -----End dump
CPU 3: Put process 1 to run queue write region=2 offset=20 value=102
CPU 3: Dispatched process 1 print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 5 Memory Dump-----
Index 20: 100
CPU 1: Put process 2 to run queue Time slot 8
CPU 1: Dispatched process 2 -----End dump
Loaded a process at input/proc/s2, PID: 4 PRI0: 120 CPU 3: Put process 1 to run queue
CPU 2: Put process 3 to run queue CPU 1: Put process 2 to run queue
CPU 2: Dispatched process 3 CPU 1: Dispatched process 1
CPU 3: Put process 1 to run queue read region=2 offset=20 value=102
CPU 0: Dispatched process 4 print_pgtbl: 0 - 1024
00000000: 80000001
Time slot 6 Loaded a process at input/proc/m0s, PID: 5 PRI0: 120
```




```
minv@DESKTOP-SLLNRQR: ~. X + v minv@DESKTOP-SLLNRQR: ~. X + v
Time slot 9
00000004: 80000000
CPU 0: Put process 4 to run queue
00000008: 80000003
CPU 3: Dispatched process 5
CPU 0: Dispatched process 6
00000012: 80000002
Memory Dump-----
Index 20: 102
-----End dump
CPU 2: Put process 3 to run queue
CPU 2: Dispatched process 3
Time slot 10
write region=3 offset=20 value=103
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Memory Dump-----
Index 20: 102
-----End dump
CPU 3: Put process 5 to run queue
CPU 3: Dispatched process 2
Time slot 11
CPU 1: Processed 1 has finished
CPU 0: Put process 6 to run queue
CPU 1: Dispatched process 4
CPU 2: Put process 3 to run queue
CPU 2: Dispatched process 6
Loaded a process at input/proc/s0, PID: 7 PRIO: 38
CPU 0: Dispatched process 5
Time slot 12
CPU 2: Put process 6 to run queue
Time slot 12
CPU 2: Put process 6 to run queue
CPU 0: Put process 5 to run queue
CPU 0: Dispatched process 6
CPU 2: Dispatched process 3
Time slot 13
CPU 3: Put process 2 to run queue
CPU 3: Dispatched process 7
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 2
Time slot 14
CPU 3: Put process 7 to run queue
CPU 1: Put process 2 to run queue
CPU 1: Dispatched process 4
CPU 3: Dispatched process 5
write region=1 offset=20 value=102
print_pgtbl: 0 - 512
00000000: 80000007
00000004: 80000006
Memory Dump-----
Index 20: 103
CPU 2: Processed 3 has finished
CPU 0: Put process 6 to run queue
CPU 0: Dispatched process 2
CPU 2: Dispatched process 7
Time slot 15
-----End dump
write region=2 offset=1000 value=1
print_pgtbl: 0 - 512
00000000: 80000007
00000004: 80000006
Memory Dump-----
Index 20: 103
Index 64: 102
```

```
minv@DESKTOP-SLLNRQR: ~. X + v minv@DESKTOP-SLLNRQR: ~. X + v
Time slot 16
-----End dump
CPU 0: Processed 2 has finished
CPU 0: Dispatched process 6
Loaded a process at input/proc/s1, PID: 8 PRIO: 0
CPU 3: Put process 5 to run queue
CPU 3: Dispatched process 5
write region=0 offset=0 value=0
print_pgtbl: 0 - 512
00000000: 80000007
00000004: 80000006
Memory Dump-----
Index 20: 103
Index 64: 102
Index 232: 1
-----End dump
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 4
Time slot 17
CPU 2: Put process 7 to run queue
CPU 2: Dispatched process 8
CPU 3: Processed 5 has finished
CPU 3: Dispatched process 7
Time slot 18
CPU 0: Put process 6 to run queue
CPU 0: Dispatched process 6
CPU 2: Put process 8 to run queue
CPU 2: Dispatched process 8
Time slot 19
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 4
CPU 3: Put process 7 to run queue
CPU 3: Dispatched process 7
Time slot 20
CPU 3: Put process 7 to run queue
CPU 3: Dispatched process 7
Time slot 20
CPU 0: Processed 6 has finished
CPU 0 stopped
CPU 2: Put process 8 to run queue
CPU 2: Dispatched process 8
CPU 1: Put process 4 to run queue
CPU 1: Dispatched process 4
Time slot 21
CPU 3: Put process 7 to run queue
CPU 3: Dispatched process 7
Time slot 22
CPU 2: Put process 8 to run queue
CPU 2: Dispatched process 8
CPU 1: Processed 4 has finished
CPU 1 stopped
Time slot 23
CPU 3: Put process 7 to run queue
CPU 3: Dispatched process 7
CPU 2: Processed 8 has finished
CPU 2 stopped
Time slot 24
Time slot 25
Time slot 26
CPU 3: Put process 7 to run queue
CPU 3: Dispatched process 7
Time slot 27
Time slot 28
CPU 3: Put process 7 to run queue
CPU 3: Dispatched process 7
Time slot 29
CPU 3: Processed 7 has finished
CPU 3 stopped
```

Gantt Chart:



Hình 4: Sơ đồ Gantt cho testcase os_1_mfq_paging_small_4K



Tài liệu tham khảo

- [1] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, Operating System Concept, Tenth Edition
- [2] Slide bài giảng môn Hệ điều hành
- [3] GeeksforGeeks, Multilevel Queue (MLQ) CPU Scheduling, truy cập từ <https://www.geeksforgeeks.org/multilevel-queue-mlq-cpu-scheduling/>
- [4] GeeksforGeeks, Virtual Memory in Operating System, truy cập từ <https://www.geeksforgeeks.org/virtual-memory-in-operating-system/>
- [5] GeeksforGeeks, Segmentation in Operating System, truy cập từ <https://www.geeksforgeeks.org/segmentation-in-operating-system/>