

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



# HỆ ĐIỀU HÀNH (CO2018)

---

Report

## SIMPLE OPERATING SYSTEM

---

GV HD:	Trần Việt Toàn	
SV thực hiện:	Võ Anh Nguyên	1914405
	Nguyễn Trung Kiên	1911441
	Phạm Nguyên Hải	1913261
	Lê Bình Đăng	1913102

TP Hồ Chí Minh, Tháng 05-06/2021



## Mục lục

1	Scheduler	2
2	Memory	5
3	Overall	8

# 1 Scheduler

Kết quả của running tests:

```
----- SCHEDULING TEST 0 -----
./os sched_0
Time slot 0
    Loaded a process at input/proc/s0, PID: 1
Time slot 1
    CPU 0: Dispatched process 1
Time slot 2
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/s1, PID: 2
Time slot 5
    CPU 0: Put process 1 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 2
Time slot 8
Time slot 9
    CPU 0: Put process 2 to run queue
    CPU 0: Dispatched process 1
Time slot 10
Time slot 11
    CPU 0: Put process 1 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 1
Time slot 14
Time slot 15
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 2
Time slot 16
    CPU 0: Processed 2 has finished
    CPU 0: Dispatched process 1
Time slot 17
Time slot 18
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 1
Time slot 19
Time slot 20
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 1
Time slot 21
Time slot 22
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 1
Time slot 23
    CPU 0: Processed 1 has finished
    CPU 0 stopped
```

MEMORY CONTENT:

NOTE: Read file output/sched\_0 to verify your result

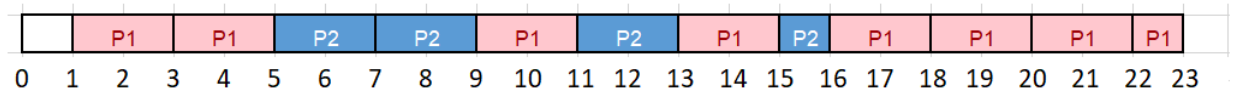
```
----- SCHEDULING TEST 1 -----  
./os sched_1  
Time slot 0  
    Loaded a process at input/proc/s0, PID: 1  
Time slot 1  
    CPU 0: Dispatched process 1  
Time slot 2  
Time slot 3  
    CPU 0: Put process 1 to run queue  
    CPU 0: Dispatched process 1  
Time slot 4  
    Loaded a process at input/proc/s1, PID: 2  
Time slot 5  
    CPU 0: Put process 1 to run queue  
    CPU 0: Dispatched process 2  
Time slot 6  
    Loaded a process at input/proc/s2, PID: 3  
Time slot 7  
    CPU 0: Put process 2 to run queue  
    CPU 0: Dispatched process 3  
    Loaded a process at input/proc/s3, PID: 4  
Time slot 8  
Time slot 9  
    CPU 0: Put process 3 to run queue  
    CPU 0: Dispatched process 4  
Time slot 10  
Time slot 11  
    CPU 0: Put process 4 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 3  
Time slot 14  
Time slot 15  
    CPU 0: Put process 3 to run queue  
    CPU 0: Dispatched process 1  
Time slot 16  
Time slot 17  
    CPU 0: Put process 1 to run queue  
    CPU 0: Dispatched process 4  
Time slot 18  
Time slot 19  
    CPU 0: Put process 4 to run queue  
    CPU 0: Dispatched process 2  
Time slot 20  
Time slot 21  
    CPU 0: Put process 2 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 1  
Time slot 24
```

```
Time slot 25
    CPU 0: Put process 1 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 2
Time slot 28
    CPU 0: Processed 2 has finished
    CPU 0: Dispatched process 3
Time slot 29
Time slot 30
    CPU 0: Put process 3 to run queue
    CPU 0: Dispatched process 1
Time slot 31
Time slot 32
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 4
Time slot 33
Time slot 34
    CPU 0: Put process 4 to run queue
    CPU 0: Dispatched process 3
Time slot 35
Time slot 36
    CPU 0: Put process 3 to run queue
    CPU 0: Dispatched process 1
Time slot 37
Time slot 38
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 4
Time slot 39
Time slot 40
    CPU 0: Put process 4 to run queue
    CPU 0: Dispatched process 3
Time slot 41
Time slot 42
    CPU 0: Processed 3 has finished
    CPU 0: Dispatched process 1
Time slot 43
Time slot 44
    CPU 0: Put process 1 to run queue
    CPU 0: Dispatched process 4
Time slot 45
    CPU 0: Processed 4 has finished
    CPU 0: Dispatched process 1
Time slot 46
    CPU 0: Processed 1 has finished
    CPU 0 stopped
```

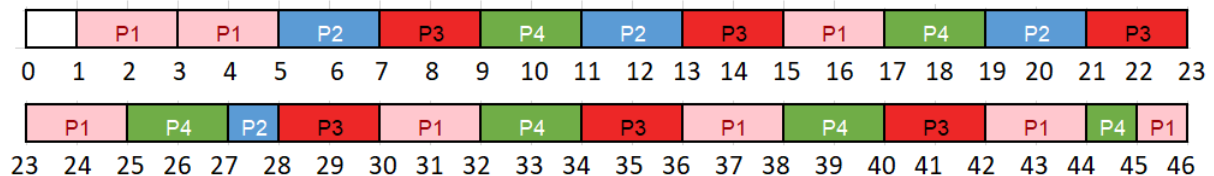
MEMORY CONTENT:

NOTE: Read `file` output/sched\_1 to verify your result

Gantt charts:



Hình 1: Scheduling test 0



Hình 2: Scheduling test 1

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

**Answer:**

Priority Feedback Queue được thiết kế dựa trên thuật toán định thời nhiều hàng chờ (multilevel feedback queue). Cụ thể, hàng đợi các quá trình sẵn sàng (ready queue) sẽ được chia thành các hàng đợi riêng biệt dựa theo các tiêu chí nào đó (độ ưu tiên và thời gian response (response time) của các quá trình). Đối với thuật toán này, ta sử dụng 2 hàng đợi **run\_queue** và **ready\_queue** với các đặc điểm như sau:

- **ready\_queue:** Hàng đợi này chứa các quá trình có độ ưu tiên thực thi cao hơn so với **run\_queue**. Cụ thể, khi CPU chuyển sang time slot kế tiếp, một quá trình trong hàng đợi này sẽ được chọn để thực thi.
- **run\_queue:** Hàng đợi này chứa các quá trình đã được thực thi một phần và đang chờ để được tiếp tục thực thi. Khi **ready\_queue** trống, toàn bộ các quá trình chứa trong **run\_queue** sẽ được đẩy vào hàng đợi **ready\_queue**.
- Cả hai hàng đợi đều là các hàng đợi sử dụng độ ưu tiên của process.

Thuật toán trên có các ưu thế hơn so với các thuật toán định thời khác (FCFS, SJF, Priority Scheduling, Round Robin, Multilevel queue) như sau:

- Linh động. Vì các tiến trình trong giải thuật PFQ có thể di chuyển giữa các hàng đợi nên tính linh hoạt rất cao.
- Tối ưu thời gian đáp ứng trung bình.
- Ngăn chặn starvation. Ở các giải thuật như FCFS, PS, SJF, MLQ nếu một tiến trình nào đó thực thi trước nhưng nếu nó không kết thúc (vì đợi kết quả từ các tiến trình có độ ưu tiên thấp hơn (PS) hoặc các tiến trình đến sau (FCFS), ...) thì các tiến trình khác sẽ không được thực thi. Ở PFQ thì điều này sẽ được ngăn chặn vì có time quantum.

## 2 Memory

Kết quả RAM Status của 2 test case m0 và m1 được biểu diễn như sau:

```
----- MEMORY MANAGEMENT TEST 0 -----
./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 000, nxt: 003)
003: 00c00-00fff - PID: 01 (idx 001, nxt: 004)
004: 01000-013ff - PID: 01 (idx 002, nxt: 005)
005: 01400-017ff - PID: 01 (idx 003, nxt: 006)
006: 01800-01bff - PID: 01 (idx 004, nxt: -01)
014: 03800-03bff - PID: 01 (idx 000, nxt: 015)
      03814: 66
015: 03c00-03fff - PID: 01 (idx 001, nxt: -01)
NOTE: Read file output/m0 to verify your result

----- MEMORY MANAGEMENT TEST 1 -----
./mem input/proc/m1
NOTE: Read file output/m1 to verify your result (your implementation should print nothing)
```

### Giải thích kết quả:

Đối với test m0, trạng thái của RAM trong quá trình thực thi được thực hiện như sau:

- alloc 13565 0: Cấp phát vùng nhớ A có kích thước  $13565/1024 \sim 14$  pages (0-13) cho process 1. Thanh ghi 0 ghi địa chỉ đầu tiên cho vùng nhớ này là 0x00000.
- alloc 1568 1: Cấp phát vùng nhớ B có kích thước  $1568/1024 \sim 2$  pages (14-15) cho process 1. Thanh ghi 1 lưu địa chỉ đầu tiên cho vùng nhớ này là 0x03800.
- free 0: Giải phóng vùng nhớ có địa chỉ bắt đầu từ thanh ghi 0. Giải phóng vùng nhớ A.
- alloc 1386 2: Cấp phát vùng nhớ C có kích thước  $1368/1024 \sim 2$  pages (0-1) cho process 1. Thanh ghi 2 lưu địa chỉ đầu tiên cho vùng nhớ C này là 0x00000.
- alloc 4564 4: Cấp phát vùng nhớ D có kích thước  $4564/1024 \sim 5$  pages (2-6) cho process 1. Thanh ghi 4 lưu địa chỉ đầu tiên cho vùng nhớ này là 0x00800.
- write 102 1 20: Ghi giá trị 102 (0x66) vào địa chỉ của thanh ghi 1 cộng thêm 20. Địa chỉ này là  $0x03800 + 0x00014 = 0x03814$ .
- write 21 2 1000: Ghi giá trị 21(0x15) vào địa chỉ của thanh ghi 2 cộng thêm 1000. Địa chỉ này là  $0x00000 + 0x003E8 = 0x003E8$ .

Đối với test1, trạng thái của RAM trong quá trình thực thi được thực hiện như sau:

- alloc 13535 0: Cấp phát vùng nhớ A có kích thước  $13535/1024 \sim 14$  pages (0-13) cho process 2. Thanh ghi 0 ghi địa chỉ đầu tiên cho vùng nhớ này là 0x00000.
- alloc 1568 1: Cấp phát vùng nhớ B có kích thước  $1568/1024 \sim 2$  pages (14-15) cho process 2. Thanh ghi 1 ghi địa chỉ đầu tiên cho vùng nhớ này là 0x00380.
- free 0: Giải phóng vùng nhớ có địa chỉ bắt đầu trong thanh ghi 0. Giải phóng vùng nhớ A.
- alloc 1386 2: Cấp phát vùng nhớ C có kích thước  $1386/1024 \sim 2$  pages (0-1) cho process 2. Thanh ghi 2 ghi địa chỉ đầu tiên cho vùng nhớ là 0x00000.
- alloc 4586 4: Cấp phát vùng nhớ D có kích thước  $4564/1024 \sim 5$  pages (2-6) cho process 2. Thanh ghi 4 ghi địa chỉ đầu tiên cho vùng nhớ này là 0x00800.
- free 2: Giải phóng vùng nhớ có địa chỉ bắt đầu trong thanh ghi 2. Giải phóng vùng nhớ C.
- free 4: Giải phóng vùng nhớ có địa chỉ bắt đầu trong thanh ghi 4. Giải phóng vùng nhớ D.
- free 1: Giải phóng vùng nhớ có địa chỉ bắt đầu trong thanh ghi 1. Giải phóng vùng nhớ B.

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

**Answer:**

Segmentation with paging là một thuật toán phân chia vùng nhớ kết hợp những ưu điểm của cả 2 giải thuật phân chia vùng nhớ segmentation và paging.

- Ưu điểm:
  - Kích thước của Page Table được giảm đi đáng kể so với Bảng phân trang trong trường hợp Single Level Paging. Lí do là vì bảng phân trang trong trường hợp này chỉ dùng đến các dữ liệu của các phân đoạn (segments) riêng biệt. Từ đó giúp giảm hao phí bộ nhớ.
  - Giảm vấn đề External Fragmentation so với việc chỉ phân đoạn do kết hợp với việc phân trang.
  - Dễ cấp phát bộ nhớ vật lí.
- Nhược điểm:
  - Internal Fragmentation vẫn tồn tại trong việc phân hoạch vùng nhớ.
  - Yêu cầu sự hỗ trợ của phần cứng.
  - Việc dịch địa chỉ từ logical address sang physical address sẽ tiến dần đến quá trình tuần tự khi số lượng các segment tăng lên, do đó làm tăng memory access time.
  - External Fragmentation vẫn xuất hiện do sự trải rộng kích thước của bảng phân trang cũng như bảng phân đoạn.
  - Khó hiện thực.



### 3 Overall

Kết quả khi thực hiện giả lập hệ điều hành đơn giản được biểu diễn như sau:

Test os\_0:

```

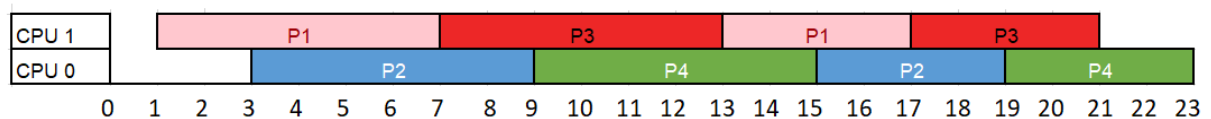
----- OS TEST 0 -----
./os os_0
Time slot 0
  Loaded a process at input/proc/p0, PID: 1
Time slot 1
  CPU 1: Dispatched process 1
Time slot 2
  Loaded a process at input/proc/p1, PID: 2
Time slot 3
  CPU 0: Dispatched process 2
  Loaded a process at input/proc/p1, PID: 3
Time slot 4
  Loaded a process at input/proc/p1, PID: 4
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 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 2
Time slot 16
Time slot 17
  CPU 1: Processed 1 has finished
  CPU 1: Dispatched process 3
Time slot 18
Time slot 19
  CPU 0: Processed 2 has finished
  CPU 0: Dispatched process 4
Time slot 20
Time slot 21
  CPU 1: Processed 3 has finished
  CPU 1 stopped

Time slot 22
Time slot 23
  CPU 0: Processed 4 has finished
  CPU 0 stopped

MEMORY CONTENT:
000: 00000-003ff - PID: 03 (idx 000, next: 001)
001: 00400-007ff - PID: 03 (idx 001, next: 002)
002: 00800-00bfff - PID: 03 (idx 002, next: 003)
003: 00c00-00fff - PID: 03 (idx 003, next: -01)
004: 01000-013ff - PID: 04 (idx 000, next: 005)
005: 01400-017ff - PID: 04 (idx 001, next: 006)
006: 01800-01bfff - PID: 04 (idx 002, next: 012)
007: 01c00-01fff - PID: 02 (idx 000, next: 008)
008: 02000-023ff - PID: 02 (idx 001, next: 009)
009: 02400-027ff - PID: 02 (idx 002, next: 010)
    025e7: 0a
010: 02800-02bfff - PID: 02 (idx 003, next: 011)
011: 02c00-02ffff - PID: 02 (idx 004, next: -01)
012: 03000-033ff - PID: 04 (idx 003, next: -01)
014: 03800-03bfff - PID: 03 (idx 000, next: 015)
015: 03c00-03fff - PID: 03 (idx 001, next: 016)
016: 04000-043ff - PID: 03 (idx 002, next: 017)
    041e7: 0a
017: 04400-047ff - PID: 03 (idx 003, next: 018)
018: 04800-04bfff - PID: 03 (idx 004, next: -01)
023: 05c00-05fff - PID: 02 (idx 000, next: 024)
024: 06000-063ff - PID: 02 (idx 001, next: 025)
025: 06400-067ff - PID: 02 (idx 002, next: 026)
026: 06800-06bfff - PID: 02 (idx 003, next: -01)
047: 0bc00-0bfff - PID: 01 (idx 000, next: -01)
    0bc14: 64
057: 0e400-0e7ff - PID: 04 (idx 000, next: 058)
058: 0e800-0ebfff - PID: 04 (idx 001, next: 059)
059: 0ec00-0efff - PID: 04 (idx 002, next: 060)
    0ede7: 0a
060: 0f000-0f3ff - PID: 04 (idx 003, next: 061)
061: 0f400-0f7ff - PID: 04 (idx 004, next: -01)
NOTE: Read file output/os_0 to verify your result

```

Hình 3: Caption



Hình 4: Overall test 0

Test os\_1:

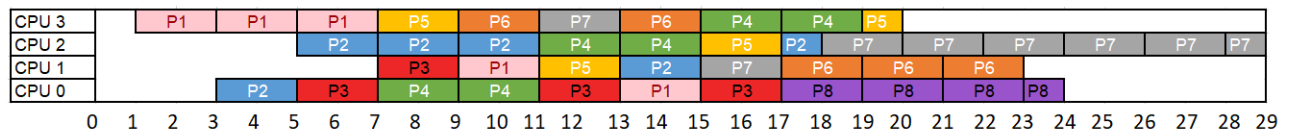
```
----- OS TEST 1 -----
./os os_1
Time slot 0
Time slot 1
    Loaded a process at input/proc/p0, PID: 1
    CPU 3: Dispatched process 1
Time slot 2
    Loaded a process at input/proc/s3, PID: 2
Time slot 3
    CPU 0: Dispatched process 2
    CPU 3: Put process 1 to run queue
    CPU 3: Dispatched process 1
Time slot 4
    Loaded a process at input/proc/m1, PID: 3
Time slot 5
    CPU 0: Put process 2 to run queue
    CPU 0: Dispatched process 3
    CPU 2: 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
Time slot 7
    CPU 0: Put process 3 to run queue
    CPU 0: Dispatched process 4
    CPU 2: Put process 2 to run queue
    CPU 2: Dispatched process 2
    Loaded a process at input/proc/m0, PID: 5
    CPU 1: Dispatched process 3
    CPU 3: Put process 1 to run queue
    CPU 3: Dispatched process 5
Time slot 8
Time slot 9
    CPU 0: Put process 4 to run queue
    Loaded a process at input/proc/pl, PID: 6
    CPU 1: Put process 3 to run queue
    CPU 1: Dispatched process 1
    CPU 3: Put process 5 to run queue
    CPU 3: Dispatched process 6
    CPU 2: Put process 2 to run queue
    CPU 2: Dispatched process 2
    CPU 0: Dispatched process 4
Time slot 10
Time slot 11
    CPU 0: Put process 4 to run queue
    CPU 1: Put process 1 to run queue
    CPU 1: Dispatched process 5
    Loaded a process at input/proc/s0, PID: 7
    CPU 3: Put process 6 to run queue
    CPU 3: Dispatched process 7
    CPU 2: Put process 2 to run queue
    CPU 2: Dispatched process 4
    CPU 0: Dispatched process 3
```

```
Time slot 12
Time slot 13
    CPU 1: Put process 5 to run queue
    CPU 1: Dispatched process 2
    CPU 0: Put process 3 to run queue
    CPU 0: Dispatched process 1
    CPU 3: Put process 7 to run queue
    CPU 3: Dispatched process 6
    CPU 2: Put process 4 to run queue
    CPU 2: Dispatched process 4
Time slot 14
Time slot 15
    CPU 1: Put process 2 to run queue
    CPU 1: Dispatched process 7
    CPU 2: Put process 4 to run queue
    CPU 0: Processed 1 has finished
    CPU 0: Dispatched process 3
    CPU 3: Put process 6 to run queue
    CPU 3: Dispatched process 4
    CPU 2: Dispatched process 5
Time slot 16
    Loaded a process at input/proc/sl, PID: 8
Time slot 17
    CPU 0: Processed 3 has finished
    CPU 0: Dispatched process 8
    CPU 2: Put process 5 to run queue
    CPU 2: Dispatched process 2
    CPU 1: Put process 7 to run queue
    CPU 1: Dispatched process 6
    CPU 3: Put process 4 to run queue
    CPU 3: Dispatched process 4
Time slot 18
    CPU 2: Processed 2 has finished
    CPU 2: Dispatched process 7
Time slot 19
    CPU 3: Processed 4 has finished
    CPU 3: Dispatched process 5
    CPU 0: Put process 8 to run queue
    CPU 0: Dispatched process 8
    CPU 1: Put process 6 to run queue
    CPU 1: Dispatched process 6
Time slot 20
    CPU 3: Processed 5 has finished
    CPU 3 stopped
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 21
    CPU 0: Put process 8 to run queue
    CPU 0: Dispatched process 8
    CPU 1: Put process 6 to run queue
    CPU 1: Dispatched process 6
Time slot 22
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 23
    CPU 0: Put process 8 to run queue
    CPU 0: Dispatched process 8
    CPU 1: Processed 6 has finished
    CPU 1 stopped
Time slot 24
    CPU 0: Processed 8 has finished
    CPU 0 stopped
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 25
Time slot 26
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 27
Time slot 28
    CPU 2: Put process 7 to run queue
    CPU 2: Dispatched process 7
Time slot 29
    CPU 2: Processed 7 has finished
    CPU 2 stopped
```

```

MEMORY CONTENT:
000: 00000-003ff - PID: 06 (idx 000, nxt: 001)
001: 00400-007ff - PID: 06 (idx 001, nxt: 007)
002: 00800-00bfff - PID: 05 (idx 000, nxt: 003)
003: 00c00-00ffff - PID: 05 (idx 001, nxt: 004)
004: 01000-013fff - PID: 05 (idx 002, nxt: 005)
005: 01400-017fff - PID: 05 (idx 003, nxt: 006)
006: 01800-01bfff - PID: 05 (idx 004, nxt: -01)
007: 01c00-01ffff - PID: 06 (idx 002, nxt: 008)
      01de7: 0a
008: 02000-023fff - PID: 06 (idx 003, nxt: 009)
009: 02400-027fff - PID: 06 (idx 004, nxt: -01)
010: 02800-02bfff - PID: 05 (idx 000, nxt: 011)
      02be8: 15
011: 02c00-02ffff - PID: 05 (idx 001, nxt: -01)
016: 04000-043fff - PID: 06 (idx 000, nxt: 017)
017: 04400-047fff - PID: 06 (idx 001, nxt: 018)
018: 04800-04bfff - PID: 06 (idx 002, nxt: 019)
019: 04c00-04ffff - PID: 06 (idx 003, nxt: -01)
021: 05400-057fff - PID: 01 (idx 000, nxt: -01)
      05414: 64
024: 06000-063fff - PID: 05 (idx 000, nxt: 025)
      06014: 66
025: 06400-067fff - PID: 05 (idx 001, nxt: -01)
NOTE: Read file output/os 1 to verify your result

```



**Hình 5:** *Overall test 1*