VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

OPERATING SYSTEMS (CO2017)

Assignment

SIMPLE OPERATING SYSTEMS

Advisor:	Dr. Le Thanh Van
Group 4:	Do thanh Vung - 2153110
	Nguyen Duc Huy - 2152088
	Mai Trong Hieu - 2152554

HO CHI MINH CITY, MAY 2023

Member list & Workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Do Thanh Vung	2153110	- Implement memory management - Asnwer questions	33.33%
2	Nguyen Duc Huy	2152088	- Implement scheduler - Answer questions	33.33%
3	Mai Trong Hieu	2152554	- Answer questions - Write report	33.33%

Contents

1 Questions regarding the implementation

1.1 Scheduler

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

- This provides a good mechanism where the relative priority of each process can be configured precisely.
- Process with higher importance can be executed first

Disadvantages:

- Lower priority processes can encounter starvation if higher priority process takes up a lot of CPU time.
- We can lose all low priority proccess if system crashes

1.2 Memory Management

1.2.1 The virtual memory mapping in each process

Q: 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? Advantages:

- Avoid internal fragmentation
- Segment Table consumes less space in comparison to Page table in paging.
- Since a complete module is loaded at once, segmentation can improves CPU ultilization
- Segmentation provides more flexibilty than paging. Segments can be of variable size, and a process can be designed in multiple segments.
- Segmentation permits sharing of memory segments between processes
- Segmentation prevents unwanted access of one process to a memory segments, which improves system stability and security

Disadvantages:

- External fragmentation can occur as processes are loaded and removed from the memory.
- Overhead is associated with keeping a segment table for each activity.
- Access time to retrieve the instruction increases as the system need to access the segment table and the main memory.
- Segmentations can be more complex to develop than paging.

1.2.2 The system physical memory

Q: What will happen if we divide the address to more than 2-levels in the paging memory manage-ment system?
sch Dividing the address into more than 2 levels in a paging memory management system can improve the flexibility and efficiency of the system. It allows for larger address spaces and finer-grained control over memory allocation. However, there are some potential drawbacks to using more than 2 levels of page tables:

- Increased overhead: Each additional level of page tables requires additional memory and computational overhead to manage. This can impact system performance and increase memory usage.
- Increased latency: With each additional level of page tables, there are more memory accesses required to access a specific page. This can increase the latency of memory accesses, which can impact overall system performance.
- Complexity: As the number of levels of page tables increases, the complexity of the paging system increases as well. This can make it more difficult to design and maintain the system.

Overall, whether to use more than 2 levels of page tables depends on the specific requirements of the system and the tradeoffs between flexibility, efficiency, and complexity.

1.2.3 Paging-based address translation scheme

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

- Reduction in memory usage
- Simpler to implement

Disadvantages:

- External fragmentation
- Requires more hardware resources
- Page tables have to be contiguois to be stores in memory

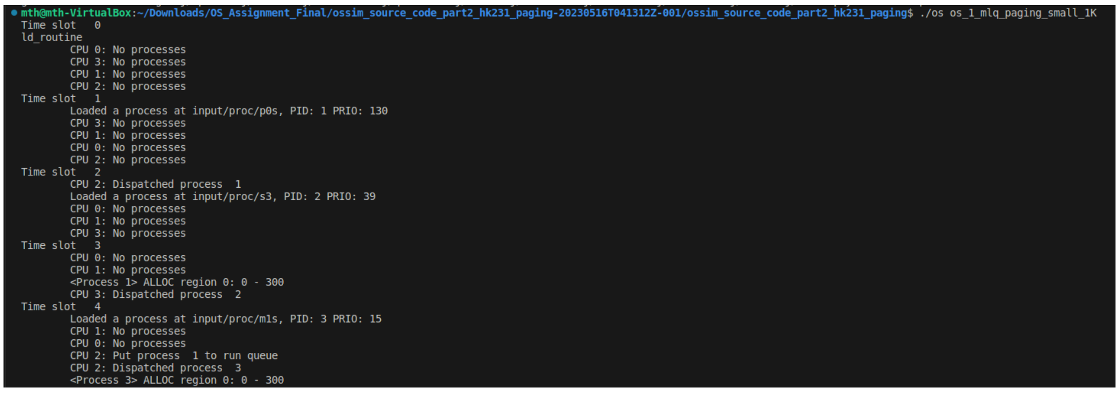
1.3 Put It All Together

Q: 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.
Importance of synchronization is operating systems:

- An operating system needs synchronization because of concurrency, interrupts, and preemption.
- If a computation can be interrupted while a shared variable is being accessed, and the interrupting computation can interact with that shared variable, then the results of the original computation can seemingly produce non-sensical results. To prevent this, we protect all accesses to shared variables with synchronization primitives, creating critical sections that cannot be concurrently accessed, thereby insuring the integrity of all computations.

- For instance: if an interupting process try to read a shared variable while it is begin written to, the result of the interrupting computation can be unpredictable. To prevent this, we need protect accesses to shared memory with synchronization primitives, resulting in critical sections which can only accessed by one process at a time, guarantee the integrity of all computation related to that critical section.

For the given test scenarios, without synchronization causes a race condition as all CPUs are trying to access the run queue as the same time.
In the figure given below, we try to run input file "os_1_mlq_paging_small_1K" without any synchronization. We can observe at time slot 2, while the process 2 should have been loaded into the run queue, none of the CPUs can access it.



```
● mth@mth-VirtualBox:~/Downloads/OS_Assignment_Final/ossim_source_code_part2_hk231_paging-20230516T041312Z-001/ossim_source_code_part2_hk231_paging$ ./os os_1_mlq_paging_small_1K
  Time slot   0
  ld_routine
          CPU 0: No processes
          CPU 3: No processes
          CPU 1: No processes
          CPU 2: No processes
  Time slot   1
          Loaded a process at input/proc/p0s, PID: 1 PRIO: 130
          CPU 3: No processes
          CPU 1: No processes
          CPU 0: No processes
          CPU 2: No processes
  Time slot   2
          CPU 2: Dispatched process  1
          Loaded a process at input/proc/s3, PID: 2 PRIO: 39
          CPU 0: No processes
          CPU 1: No processes
          CPU 3: No processes
  Time slot   3
          CPU 0: No processes
          CPU 1: No processes
          <Process 1> ALLOC region 0: 0 - 300
          CPU 3: Dispatched process  2
  Time slot   4
          Loaded a process at input/proc/m1s, PID: 3 PRIO: 15
          CPU 1: No processes
          CPU 0: No processes
          CPU 2: Put process  1 to run queue
          CPU 2: Dispatched process  3
          <Process 3> ALLOC region 0: 0 - 300
```

Figure 1: Race condition

2 Result intepreting

2.1 Scheduling

In this section, we will use a Gnatt diagram to describe how processes are executed by the CPU

Note: In order for the program to run input file "sched", "sched", "sched" sucessfully, we need to comment out the following block of code in the os.c file.

```
    #ifdef MM_FIXED_MEMSZ
            memramsz = 0x100000;
            memswpsz[0] = 0x1000000;
            for (sit = 1; sit < PAGING_MAX_MMSWP; sit++)
                    memswpsz[sit] = 0;
    #else
            fscanf(file, "%d\n", &memramsz);
            for (sit = 0; sit < PAGING_MAX_MMSWP; sit++)
                    fscanf(file, "%d", &(memswpsz[sit]));

            fscanf(file, "\n");
```

```
#endif
```

2.1.1 Sched

Output:

```
Time slot 0
        CPU 0: No processes
ld_routine
        CPU 1: No processes
        Loaded a process at input/proc/p1s, PID: 1 PRIO: 1
        Loaded a process at input/proc/p1s, PID: 2 PRIO: 0
        CPU 0: Dispatched process 2
        CPU 1: Dispatched process 1
Time slot 1
        Loaded a process at input/proc/p1s, PID: 3 PRIO: 0
Time slot 2
Time slot 3
Time slot 4
        CPU 1: Put process 1 to run queue
        CPU 1: Dispatched process 3
        CPU 0: Put process 2 to run queue
        CPU 0: Dispatched process 2
Time slot 5
Time slot 6
Time slot 7
Time slot 8
        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
Time slot 9
Time slot 10
Time slot 11
        CPU 0: Processed 2 has finished
        CPU 0: Dispatched process 1
Time slot 12
Time slot 13
        CPU 1: Put process 3 to run queue
        CPU 1: Dispatched process 3
Time slot 14
Time slot 15
        CPU 1: Processed 3 has finished
        CPU 1 stopped
        CPU 0: Put process 1 to run queue
        CPU 0: Dispatched process 1
Time slot 16
```

```
    Time slot 17
            CPU 0: Processed 1 has finished
            CPU 0 stopped
```

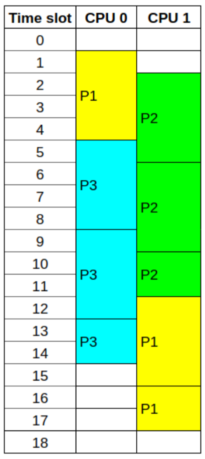
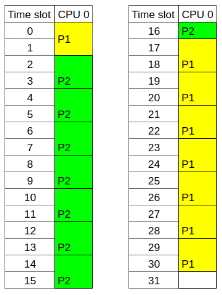


Figure 2: Gnatt diagram of sched

2.1.2 Sched 0

```
Time slot 0
ld_routine
        CPU 0: No processes
        Loaded a process at input/proc/s0, PID: 1 PRIO: 4
Time slot 1
        CPU 0: Dispatched process 1
        Loaded a process at input/proc/s0, PID: 2 PRIO: 0
Time slot 2
Time slot 3
        CPU 0: Put process 1 to run queue
        CPU 0: Dispatched process 2
Time slot 4
Time slot 5
        CPU 0: Put process 2 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 2
Time slot 10
Time slot 11
        CPU 0: Put process 2 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 2
Time slot 14
Time slot 15
        CPU 0: Put process 2 to run queue
        CPU 0: Dispatched process 2
Time slot 16
Time slot 17
        CPU 0: Put process 2 to run queue
        CPU 0: Dispatched process 2
Time slot 18
        CPU 0: Processed 2 has finished
        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
     Time slot 24
            CPU 0: Put process 1 to run queue
            CPU 0: Dispatched process 1
     Time slot 25
     Time slot 26
            CPU 0: Put process 1 to run queue
            CPU 0: Dispatched process 1
     Time slot 27
     Time slot 28
            CPU 0: Put process 1 to run queue
            CPU 0: Dispatched process 1
     Time slot 29
     Time slot 30
            CPU 0: Put process 1 to run queue
            CPU 0: Dispatched process 1
     Time slot 31
            CPU 0: Processed 1 has finished
            CPU 0 stopped
```



Time slot	CPU 0	Time slot	CPU 0
0	P1	16	P2
1		17	
2		18	P1
3	P2	19	
4		20	P1
5	P2	21	
6		22	P1
7	P2	23	
8		24	P1
9	P2	25	
10		26	P1
11	P2	27	
12		28	P1
13	P2	29	
14		30	P1
15	P2	31	

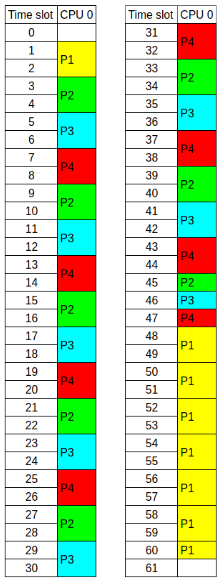
Figure 3: Gnatt diagram of sched 0

2.1.3 Sched 1

```
Time slot 0
ld_routine
        CPU 0: No processes
        Loaded a process at input/proc/s0, PID: 1 PRIO: 4
Time slot 1
        CPU 0: Dispatched process 1
        Loaded a process at input/proc/s0, PID: 2 PRIO: 0
Time slot 2
        Loaded a process at input/proc/s0, PID: 3 PRIO: 0
Time slot 3
        CPU 0: Put process 1 to run queue
        CPU 0: Dispatched process 2
        Loaded a process at input/proc/s0, PID: 4 PRIO: 0
Time slot 4
Time slot 5
        CPU 0: Put process 2 to run queue
        CPU 0: Dispatched process 3
Time slot 6
Time slot 7
        CPU 0: Put process 3 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 2
Time slot 10
Time slot 11
        CPU 0: Put process 2 to run queue
        CPU 0: Dispatched process 3
Time slot 12
Time slot 13
        CPU 0: Put process 3 to run queue
        CPU 0: Dispatched process 4
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 0: Put process 2 to run queue
        CPU 0: Dispatched process 3
Time slot 18
Time slot 19
        CPU 0: Put process 3 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 2
    Time slot 22
    Time slot 23
            CPU 0: Put process 2 to run queue
            CPU 0: Dispatched process 3
    Time slot 24
    Time slot 25
            CPU 0: Put process 3 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
    Time slot 29
            CPU 0: Put process 2 to run queue
            CPU 0: Dispatched process 3
    Time slot 30
    Time slot 31
            CPU 0: Put process 3 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 2
    Time slot 34
    Time slot 35
            CPU 0: Put process 2 to run queue
            CPU 0: Dispatched process 3
    Time slot 36
    Time slot 37
            CPU 0: Put process 3 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 2
    Time slot 40
    Time slot 41
            CPU 0: Put process 2 to run queue
            CPU 0: Dispatched process 3
    Time slot 42
    Time slot 43
            CPU 0: Put process 3 to run queue
            CPU 0: Dispatched process 4
    Time slot 44
    Time slot 45
```

```
            CPU 0: Put process 4 to run queue
            CPU 0: Dispatched process 2
     Time slot 46
            CPU 0: Processed 2 has finished
            CPU 0: Dispatched process 3
     Time slot 47
            CPU 0: Processed 3 has finished
            CPU 0: Dispatched process 4
     Time slot 48
            CPU 0: Processed 4 has finished
            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
```



Time slot	CPU 0
0	
1	P1
2	
3	P2
4	
5	P3
6	
7	P4
8	
9	P2
10	
11	P3
12	
13	P4
14	
15	P2
16	
17	P3
18	
19	P4
20	
21	P2
22	
23	P3
24	
25	P4
26	
27	P2
28	
29	P3
30	
31	P4
32	
33	P2
34	
35	P3
36	
37	P4
38	
39	P2
40	
41	P3
42	
43	P4
44	
45	P2
46	P3
47	P4
48	P1
49	
50	P1
51	
52	P1
53	
54	P1
55	
56	P1
57	
58	P1
59	
60	P1
61	

Figure 4: Gnatt diagram of sched 1

2.2 Memory

In this section, we will track the RAM status throughout the running duration of the program using the input file: ”alloc”

Input file:

```
6 2 4
1048576 16777216 0 0 0
0 p0s 0
2 p1s 15
```

Output:

```
Time slot 0
ld_routine
        CPU 0: No processes
        CPU 1: No processes
        CPU 2: No processes
        CPU 3: No processes
Time slot 1
        Loaded a process at input/proc/p0s, PID: 1 PRIO: 130
        CPU 0: No processes
        CPU 3: Dispatched process 1
        CPU 2: No processes
        CPU 1: No processes
Time slot 2
        CPU 1: No processes
        Loaded a process at input/proc/s3, PID: 2 PRIO: 39
        <Process 1> ALLOC region 0: 0 - 300
        CPU 0: Dispatched process 2
        CPU 2: No processes
Time slot 3
        CPU 1: No processes
        CPU 3: Put process 1 to run queue
        CPU 3: Dispatched process 1
        <Process 1> ALLOC region 4: 512 - 812
        CPU 2: No processes
        Loaded a process at input/proc/m1s, PID: 3 PRIO: 15
Time slot 4
        CPU 0: Put process 2 to run queue
        CPU 0: Dispatched process 3
        <Process 3> ALLOC region 0: 0 - 300
        CPU 2: No processes
        CPU 1: Dispatched process 2
        <Process 1> FREE region 0
Time slot 5
        <Process 3> ALLOC region 1: 512 - 612
        CPU 2: No processes
        CPU 3: Put process 1 to run queue
        CPU 3: Dispatched process 1
        <Process 1> ALLOC region 1: 200 - 300
Time slot 6
        CPU 0: Put process 3 to run queue
        CPU 0: Dispatched process 3
        <Process 3> FREE region 0
        CPU 2: No processes
        Loaded a process at input/proc/s2, PID: 4 PRIO: 120
        CPU 1: Put process 2 to run queue
        CPU 1: Dispatched process 2
write region=1 offset=20 value=100
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 7
        <Process 3> ALLOC region 2: 200 - 300
        CPU 2: Dispatched process 4
        CPU 3: Put process 1 to run queue
        CPU 3: Dispatched process 1
read region=1 offset=20 value=100
print_pgtbl: 0 - 1024
```

```
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
        Loaded a process at input/proc/m0s, PID: 5 PRIO: 120
Time slot 8
        CPU 0: Put process 3 to run queue
        CPU 0: Dispatched process 3
        <Process 3> FREE region 2
        CPU 1: Put process 2 to run queue
        CPU 1: Dispatched process 2
write region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 9
        <Process 3> FREE region 1
        CPU 2: Put process 4 to run queue
        CPU 2: Dispatched process 5
        <Process 5> ALLOC region 0: 0 - 300
        Loaded a process at input/proc/p1s, PID: 6 PRIO: 15
        CPU 3: Put process 1 to run queue
        CPU 3: Dispatched process 6
Time slot 10
        CPU 0: Put process 3 to run queue
        CPU 0: Dispatched process 3
        <Process 3> FREE region 0
        <Process 5> ALLOC region 1: 512 - 612
        CPU 1: Put process 2 to run queue
        CPU 1: Dispatched process 2
Time slot 11
        <Process 3> FREE region 0
        CPU 2: Put process 5 to run queue
        CPU 2: Dispatched process 4
        Loaded a process at input/proc/s0, PID: 7 PRIO: 38
        CPU 3: Put process 6 to run queue
        CPU 3: Dispatched process 6
Time slot 12
        CPU 0: Processed 3 has finished
        CPU 0: Dispatched process 7
        CPU 1: Put process 2 to run queue
        CPU 1: Dispatched process 2
Time slot 13
        CPU 2: Put process 4 to run queue
        CPU 2: Dispatched process 5
        <Process 5> FREE region 0
        CPU 1: Processed 2 has finished
        CPU 1: Dispatched process 4
        CPU 3: Put process 6 to run queue
        CPU 3: Dispatched process 6
Time slot 14
        CPU 0: Put process 7 to run queue
        CPU 0: Dispatched process 7
        <Process 5> ALLOC region 2: 200 - 300
Time slot 15
        CPU 1: Put process 4 to run queue
        CPU 1: Dispatched process 4
        CPU 3: Put process 6 to run queue
        CPU 3: Dispatched process 6
        CPU 2: Put process 5 to run queue
        CPU 2: Dispatched process 5
write region=1 offset=20 value=102
print_pgtbl: 0 - 768
00000000: 80000000
00000004: 80000002
00000008: 80000001
Time slot 16
        Loaded a process at input/proc/s1, PID: 8 PRIO: 0
        CPU 0: Put process 7 to run queue
        CPU 0: Dispatched process 8
write region=2 offset=1000 value=1
print_pgtbl: 0 - 768
00000000: 80000000
00000004: 80000002
00000008: 80000001
Time slot 17
        CPU 1: Put process 4 to run queue
        CPU 1: Dispatched process 7
        CPU 3: Put process 6 to run queue
        CPU 3: Dispatched process 6
        CPU 2: Put process 5 to run queue
        CPU 2: Dispatched process 4
Time slot 18
        CPU 0: Put process 8 to run queue
        CPU 0: Dispatched process 8
Time slot 19
        CPU 1: Put process 7 to run queue
        CPU 1: Dispatched process 7
        CPU 2: Put process 4 to run queue
        CPU 2: Dispatched process 5
write region=0 offset=0 value=0
print_pgtbl: 0 - 768
00000000: 80000000
00000004: 80000002
00000008: c0000000
        CPU 3: Processed 6 has finished
```

```
            CPU 3: Dispatched process 4
    Time slot 20
            CPU 0: Put process 8 to run queue
            CPU 0: Dispatched process 8
            CPU 2: Processed 5 has finished
            CPU 2: Dispatched process 1
    read region=2 offset=20 value=102
    print_pgtbl: 0 - 1024
    00000000: 80000001
    00000004: 80000000
    00000008: 80000003
    00000012: 80000002
    Time slot 21
            CPU 1: Put process 7 to run queue
            CPU 1: Dispatched process 7
            CPU 3: Processed 4 has finished
            CPU 3 stopped
    write region=3 offset=20 value=103
    print_pgtbl: 0 - 1024
    00000000: 80000001
    00000004: 80000000
    00000008: 80000003
    00000012: 80000002
    Time slot 22
            CPU 0: Put process 8 to run queue
            CPU 0: Dispatched process 8
            CPU 2: Processed 1 has finished
            CPU 2 stopped
    Time slot 23
            CPU 0: Processed 8 has finished
            CPU 0 stopped
            CPU 1: Put process 7 to run queue
            CPU 1: Dispatched process 7
    Time slot 24
    Time slot 25
            CPU 1: Put process 7 to run queue
            CPU 1: Dispatched process 7
    Time slot 26
    Time slot 27
            CPU 1: Put process 7 to run queue
            CPU 1: Dispatched process 7
    Time slot 28
            CPU 1: Processed 7 has finished
            CPU 1 stopped
```

Now, we can analyze the RAM by focusing on the ALLOC, FREE, WRITE and READ actions. We will only need to look at time slot 2 − > 17 to see all features of the memory management system in action.

- **Time slot 2**
  Process 1: Alloc region 0, size 300
  Result: Alloc region 0: 0 - 300
  Explanation: No suitable region in the free region list, create new region 0 - 512 (2 pages).
  Free region list: 300 - 512

  Total ram frames: 2

- **Time slot 3**
  Process 1: Alloc region 4, size 300
  Result: Alloc region 4: 512 - 812
  Explanation: No suitable region in the free region list, create new region 512 - 1024 (2 pages).
  Free region list: 300 - 512, 812 - 1024

  Total ram frames: 4

- **Time slot 4**
  Process 3: Alloc region 0, size 300
  Result: Alloc region 0: 0 - 300
  Explanation: 0 - 300 will now be added to free region list.
  Free region list (P3): 300 - 512

  Process 1: free 0
  Result: Free region 0
  Explanation: 0 - 300 will now be added to free region list.
  Free region list (P1): 300 - 512, 0 - 300

  Total ram frames: 6

- **Time slot 5**
  Process 3: Alloc region 1, size 100
  Result: Alloc region 1: 512 - 612
  Explanation: No suitable region in the free region list, create new region 512 - 768 (1 pages).
  Free region list (P3): 300 - 512, 612 - 768

  Process 1: Alloc region 1, size 100
  Result: Alloc region 1: 200 - 300
  Explanation: 0 - 300 will now be added to free region list.
  Free region list (P1): 300 - 512, 0 - 200

  Total ram frames: 7

- **Time slot 6**
  Process 3: free 0
  Result: Free region 0

Free region list (P3): 0 - 300, 300 - 512, 612 - 768

Process 1: write region 0, offset 20, value 100
Result: write to address 220 of page 0
Explanation: Page 0 is currently in ram, can be accessed directly.
Free region list (P1): 300 - 512, 0 - 200

Total ram frames: 7

- **Time slot 7**
  Process 3: Alloc region 2, size 100
  Result: Alloc region 2: 200 - 300
  Explanation: Suitable region found in free list.
  Free region list (P3): 0 - 200, 300 - 512, 612 - 768

  Process 1: Read region 1, offset 20
  Result: value = 100
  Explanation: Address is 220, page 0, access page directly through ram.
  Free region list (P1): 300 - 512, 0 - 200

  Total ram frames: 7

2.3 Overall

In this section, we will use the input file: ”os_0_mlq_paging” to test all features of the OS

Note: before starting the program, we should uncomment the block of code previously mentioned in section 2.1

Input file:

```
6 2 4
1048576 16777216 0 0 0
0 p0s 0
2 p1s 15
```

Output:

```
Time slot 0
ld_routine
        CPU 0: No processes
        CPU 1: No processes
        Loaded a process at input/proc/p0s, PID: 1 PRIO: 0
Time slot 1
        CPU 0: Dispatched process 1
        CPU 1: No processes
Time slot 2
```

```
        Loaded a process at input/proc/p1s, PID: 2 PRIO: 15
        CPU 1: Dispatched process 2
        <Process 1> ALLOC region 0: 0 - 300
Time slot 3
        Loaded a process at input/proc/p1s, PID: 3 PRIO: 0
        <Process 1> ALLOC region 4: 512 - 812
Time slot 4
        Loaded a process at input/proc/p1s, PID: 4 PRIO: 0
        <Process 1> FREE region 0
Time slot 5
        <Process 1> ALLOC region 1: 200 - 300
Time slot 6
write region=1 offset=20 value=100
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 7
        CPU 0: Put process 1 to run queue
        CPU 0: Dispatched process 3
Time slot 8
        CPU 1: Put process 2 to run queue
        CPU 1: Dispatched process 4
Time slot 9
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
Time slot 14
        CPU 1: Put process 4 to run queue
        CPU 1: Dispatched process 3
write region=2 offset=20 value=102
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 15
read region=2 offset=20 value=102
print_pgtbl: 0 - 1024
```

```
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 16
write region=3 offset=20 value=103
print_pgtbl: 0 - 1024
00000000: 80000001
00000004: 80000000
00000008: 80000003
00000012: 80000002
Time slot 17
       CPU 0: Processed 1 has finished
       CPU 0: Dispatched process 4
Time slot 18
       CPU 1: Processed 3 has finished
       CPU 1: Dispatched process 2
Time slot 19
Time slot 20
Time slot 21
       CPU 0: Processed 4 has finished
       CPU 0 stopped
Time slot 22
       CPU 1: Processed 2 has finished
       CPU 1 stopped
```