**Bài 1:**

|  |  |  |
| --- | --- | --- |
| Segment | Base | Limit |
| 0 | 219 | 600 |
| 1 | 2300 | 14 |
| 2 | 90 | 100 |
| 3 | 1327 | 580 |
| 4 | 1952 | 96 |

1. 0, 701 => invalid
2. 1, 8 => 2300+8=2308
3. 2, 100 => 90+100=190
4. 3, 429 => 1327+429=1756
5. 4, 78 => 1952 + 78=2030

**Bài 2:**

Với thuật toán First-fit: Với thuật toán Best-fit: Với thuật toán Worst-fit

|  |
| --- |
| P3 (417KB) |
|  |
| 100KB |
| P1(112KB) |
|  |
| P4 (426KB) |
|  |
| P2 (212KB) |
|  |

|  |
| --- |
| P1(112KB) |
| P2 (212KB) |
| 176KB |
| 100KB |
| 200KB |
| P3 (417KB) |
| 183KB |
| 300KB |

|  |
| --- |
| P2 (212KB) |
|  |
| 100KB |
|  |
|  |
| P1(112KB) |
| P3 (417KB) |
|  |
| 300KB |

Ưu điểm: Nhanh

Nhược điểm: Không đủ không gian liên tục để chứa P4 (426KB)=> Phân mảnh ngoại vi (external fragmentation)

Ưu điểm: Nhanh

Nhược điểm: Không đủ không gian liên tục để chứa P4 (426KB

Ưu điểm: Đủ không gian để chứa tất cả các tiến trình

Nhược điểm: phải tìm kiếm hêt bộ nhớ => vấn đề tốc độ

* Kết luận: Thuật toán Best-fit là lựa chọn tôt nhất, mặc dù phải tìm kiếm hết bộ nhớ nhưng có thể tìm được không gian phù hợp cho cả 4 tiến trình

**Câu 3:**

1. **FIFO:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
| 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 3 | 3 |
|  |  | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
|  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| \* | \* | \* | \* |  |  | \* | \* | \* | \* |  | \* | \* | \* |  | \* | \* |  | \* |  |

* Số page faults: 14
* Tần suất xảy ra page fault: 0.7
* EAT=(1-p)\*tm+p\*tp=0,3\*102+0,7\*192=30,6+134,4=165 ns

1. **Optimal**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7 | 7 | 7 | 7 | 1 | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  |  | 3 | 3 | 3 | 3 | 3 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
|  |  |  | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| \* | \* | \* | \* |  |  | \* | \* |  |  |  | \* | \* |  |  |  | \* |  |  |  |

* Số page faults: 9
* Tần suất xảy ra page fault: 0,45
* EAT=(1-p)\*tm+p\*tp=0,55\*102+0,45\*192=56,1+86,4=142,5 ns

1. **LRU:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  |  | 3 | 3 | 3 | 3 | 5 | 6 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
|  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 1 | 1 | 1 | 1 |
| \* | \* | \* | \* |  |  | \* | \* |  |  |  | \* | \* | \* |  |  | \* |  |  |  |

* Số page fault: 10
* Tần suất xảy ra page fault: 0,5
* EAT=(1-p)\*tm+p\*tp=0,5\*102+0,5\*192=51+96=147 ns

1. **Second change:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | | 4 | | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 |
| Frame  1 | 1 | 1 | 1 | | 1 | | 1 | 1 | 5 | 5 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 1 |
| Frame  2 |  | 2 | 2 | | 2 | | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 |
| Frame  3 |  |  | 3 | | 3 | | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 |
| Frame  4 |  |  |  | | 4 | | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| Fifo | 1 | 12 | 123 | | 1234 | | 1234 | 1234 | 2345 | 3456 | 4562 | 5621 | 5621 | 6213 | 2137 | 1376 | 1376 | 3762 | 7621 |
| Refe  rence bit | 1 | 12 | 123 | | 1234 | | 1234 | 1234 | 5 | 56 | 562 | 5621 | 5621 | 3 | 37 | 376 | 376 | 3762 | 1 |
|  | \* | \* | \* | | \* | |  |  | \* | \* | \* | \* |  | \* | \* | \* |  | \* | \* |
|  | 2 | 3 | | 6 | |
| Frame  1 | 1 | 1 | | 1 | |
| Frame  2 | 7 | 3 | | 3 | |
| Frame  3 | 6 | 6 | | 6 | |
| Frame  4 | 2 | 2 | | 2 | |
| Fifo | 7621 | 6213 | | 6213 | |
| Refe  rence bit | 12 | 123 | | 1236 | |
|  |  | \* | |  | |

* Số page faults: 14
* Tần suất xảy ra page fault: 0.7
* EAT=(1-p)\*tm+p\*tp=0,3\*102+0,7\*192=30,6+134,4=165 ns

**Câu 4:**

Page size=1KB

Process P => 8 pages

32 frame

1. Calculate the number of bits used for storing a physical address

* Số frame là 32 = 25 => Dùng 5 bit để lưu no of Frame
* Kích thước frame = kích thước page =1KB =2^10 => Dùng 10 bit để lưu số Offset
* SỐ lượng bit cần dùng để lưu bộ nhớ vật lý: 10+5=15 bíts

1. Calculate the size of virtual memory space of the program P:

Kích thước bộ nhớ ảo = Page size \* Number of pages = 1KB \* 8 pages =8 KB

1. Calculate the size of physical memory:
2. Kích thước bộ nhớ vật lý = Frame size \* Number of frames = 1KB \* 32 frames =32 KB

**Câu 5:**

48 bit logical address

Page size = 8KB=

Byte-addressable memory: 320MB

1. Give the number of frames in the physical memory:

No.frames=320MB/8KB=10\*225/213=10\*212=40960

1. Give the maximum number of pages in logical address space:

1 page =213 => Dùng 13 bit để lưu offsets

* Số bits dùng để lưu số page: 35 bits => Max no,pages: 235

1. Convert the logical addresses 1892, 15296 and 20300 to addresses <p,d>

1892 => p=1, d=1892

15296 => p=2, d=7104

20300 => p=3, d=3916

**Câu 6:**

1. Số lượng bit dùng để lưu offset là: 32-9-11=12 bits

* Size of page: 212=4 KB

1. Give the number of frames in the memory:

No. frames=10GB/4KB=10\*230/4\*210=2560000

1. What is the maximum size of process space supported in thí system?

Số lượng bits dành cho page number: 32-12=20 bits

Kích thước lớn nhất cho không gian tiến trình: 220 byte=1MB

1. If loading the process P1 of 2.8GB into the system, may we suffer from the fragmentation problem? Justify your answer

Giả sử tiến trình P1 đi vào hệ thống, no. Pages = 2,8\*230/4\*210=734003 page

1. Trong hệ thống paging, phân mảnh có thể là internal(trong 1 page) hoặc external (giữa các page).

**Câu 7:**

EAT= h \* (tc+tm) +(1-h)\*(tc+4\*tm)=0,75 \* (25+132) + 0,25\*(25+4\*132)=256 ns

**Câu 8:**

EAT= h \* (tc+tm) +(1-h)\*(tc+2\*tm)

=> 225=h \*(25+150)+(1-h)\*(25+2\*150) ⬄ 225=175h+325(1-h)

⬄ h=0,67=67%