1. Consider a logical address space of 64 pages of 1,024 words each, mapped onto a physical memory of 32 frames.
   1. How many bits are there in the logical address?

Size of logical address space = 2m = #of pages\*page size

=64\*1024 = 210 \* 26 = 216 => m=16

Vậy có 16 bit in the logical address.

* 1. How many bits are there in the physical address?

Size of physical address space = 2n = #of pages\*page size

=32\*1024 = 25 \* 210 = 215 => n=15

Vậy có 15 bit in the physical address.

1. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)?

*First-fit:*

115KB/300KB(185KB); 500KB/600KB(100KB); 358KB/750KB(392KB); 200KB/350KB(150KB)

375KB/392KB(17)

*Best-fit:*

115KB/125KB(10); 500KB/600KB(100); 358KB/750KB(392); 200KB/200KB; 375KB/392KB(17)

*Worst-fit:*

115KB/750KB(635); 500KB/635KB(135); 358KB/600KB(242); 200KB/350KB(150); 375 chờ

1. Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

Cách 1: 1 Kb = 1024 Byte

* 1. 3085 div 1024 = 3 page, 3085 mod 1024 = 13 offset
  2. 42095 div 1024 = 41 page, 42095 mod 1024 = 111 offset
  3. 215201 div 1024 = 210 page, 215201 mod 1024 = 161 offset
  4. 650000 div 1024 = 634 page, 650000 mod 1024 = 784 offset
  5. 2000001 div 1024 = 1953 page, 2000001 mod 1024 = 129 offset

Cách 2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Logical address  (decimal) | Logical address  (binary) | Page # 22 bit  (binary) | Page # 10 bit  (binary) | Page#  (decimal) | Offset  (decimal) |
| 3085 | 0000 1100 0000 1101 | 0000 11 | 000000 1101 | 3 | 13 |
| 42095 | 1010 0100 0110 1111 | 1010 01 | 00 0110 1111 | 41 | 111 |
| 215201 | 0000 0000 0000 0011 0100 1000 1010 0001 | 0000 0000 0000 0011 0100 10 | 00 1010 0001 | 210 | 161 |
| 650000 | 0000 0000 0000 1001 1110 1011 0001 0000 | 0000 0000 0000 1001 1110 10 | 11 0001 0000 | 634 | 784 |
| 2000001 | 0000 0000 0001 1110 1000 0100 1000 0001 | 0000 0000 0001 1110 1000 01 | 00 1000 0001 | 1953 | 129 |

1. Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a **physical memory of 64 frames.**
   1. **How many bits are required in the logical address?**

**256\*4\*1024 = 28 \* 22 \* 210 = 220. Vậy có 20 bit.**

* 1. **How many bits are required in the physical address?**

**64\*4\*1024 = 26 \* 22 \* 210 = 220. Vậy có 28 bit**

1. **Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in the single-level page table?**

Size of logical address space = 2m = #of pages\*page size

=232 = #of pages\* 4\*1024 = #of pages\* 212

=># **of pages =**232 / 212 = 220 => Vậy cần 20 bit

1. Consider the following segment table:

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

What are the physical addresses for the following logical addresses?

* 1. 0,430 => 430<600 – True >> Physical address = 219+430 = 649
  2. 1,10 => 10<14 – True >> Physical address = 2300+10 = 2310
  3. 2,500 => 500>100 – False >> trap; addressing error
  4. 3,400 => 400<580 –True >> Physical address = 1327+400 = 1727
  5. 4,112 => 112>96 - False>> trap; addressing error