

**Assignment 2**

**Name: Muhammad Shaheer**

**Roll no: 20p-0480**

**Subject: Operating Systems**

Q1: Why are page sizes always powers of 2?

Answer: Many early computer architectures and operating systems used page sizes that were powers of 2.Computers operate internally in binary, which is a base-2 number system. Using a page size that is a power of 2 simplifies the memory management process. Using powers of 2 as page sizes ensures that data structures within memory can be aligned properly. Page size is always power of 2 because if page size is power of 2 then it is easier to break the address into two parts. Recall that paging is implemented by breaking up an address into a page and off set number. It is most efficient to break the address into X page bits and Y off set bits, rather than perform arithmetic on the address to calculate the page number and off set. Because each bit position represents a power of 2, splitting an address between bits results in a page size that is a power of 2.

Q2: Consider a logical address space of 64 pages of 1,024 words each, mapped onto a physical

Memory of 32 frames.

a. How many bits are there in the logical address?

b. How many bits are there in the physical address?

Answer: To determine the number of bits required for the logical address, we need to consider the size of the logical address space.

a) Logical memory = 64 pages

= 2⁶ pages

Size of each word = 1024 = 2¹⁰

Hence total logical memory = 2⁶ x 2¹⁰ = 2¹⁶

Hence the logical address will have 16 bits.

b) The physical address determines the actual location of the data in physical memory. Since there are 32 frames in physical memory, we need 5 bits to address each frame uniquely.

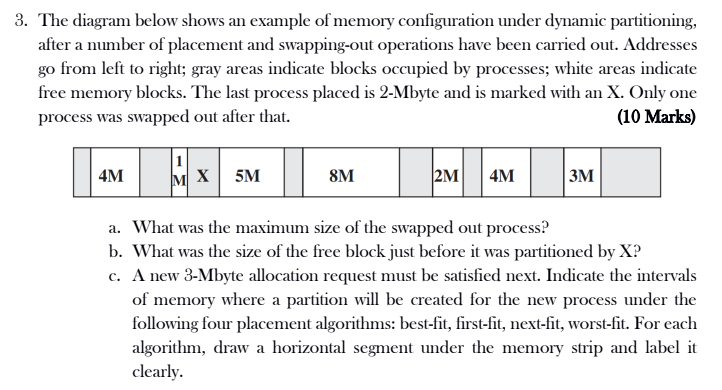
Physical memory = 32 frames

= 2⁵ frames

Size of each word = 1024 = 2¹⁰

Hence total physical size = 2⁵ x 2¹⁰ = 2¹⁵

Hence there will be 15 bits in the physical address



Ans:

A: The maximum size of the swapped out process is 4 Mbytes. This is because the largest contiguous block of free memory is 4 Mbytes, and the swapped out process must have been smaller than or equal to this size.

B: The size of the free block just before it was partitioned by X is 2 Mbytes. This is because the block occupied by X is 2 Mbytes, and the next largest contiguous block of free memory is 2 Mbytes.

C:

The following table shows the intervals of memory where a partition will be created for the new process under the following four placement algorithms:

|  |  |
| --- | --- |
| Placement Algorithm | Memory Intervals |
| Best-fit | [0, 3] |
| First-fit | [2, 5] |
| Next-fit | [2, 5] |
| Worst-fit | [0, 7] |

Under best-fit, the 3-Mbyte process is allocated to the smallest contiguous block of free memory, which is [0, 3]. Under first-fit, the 3-Mbyte process is allocated to the first contiguous block of free memory that is large enough, which is [2, 5]. Under next-fit, the 3-Mbyte process is allocated to the next block of free memory that is large enough, which is also [2, 5]. Under worst-fit, the 3-Mbyte process is allocated to the largest contiguous block of free memory, which is [0, 7].

Here is a diagram of the memory configuration under each placement algorithm:

[0] [1] [2] [3] [4] [5] [6] [7]

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | X | | | |

| | | | | | | |

Best-fit:

[0] [1] [2] [3] [4] [5] [6] [7]

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | X | | | |

First-fit:

[0] [1] [2] [3] [4] [5] [6] [7]

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | X | | | | |

Next-fit:

[0] [1] [2] [3] [4] [5] [6] [7]

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | | | | | |

| | | X | | | | |

Worst-fit:

[0] [1] [2] [3] [4] [5] [6] [7]

| | | | | | | |

| | | | | | | |

| | | | | |