Deliverables Lesson 6

Sistemas Operativos

Grado en Ingeniería Informática

Departamento de Ingeniería Informática

Universidad de Cádiz

Deliverable 6.1

Consider a system with a physical memory of 512 MiB. The operating system occupies 512 KiB. Which will be the maximum size of a process in each of the following types of memory allocation schemes?

1. A fixed partitioning system where all partitions have the same size: 256 KiB.
2. A dynamic partitioning system.
3. A buddy system with a minimum partition size of 64 KiB.
4. A virtual memory system with 32-bit logical addresses.

Deliverable 6.1 Solution

1. *The size of the partition limits the size of the process. Thus, the maximum size of the process is 256 KiB.*
2. *The operating system occupies 512 KiB, therefore we have (512 MiB - 512 KiB) available for processes. This is the maximum size of a process.*
3. *To allocate memory to the operating system, the partition of 512 MiB will be divided in two partitions of 256 MiB. One of these will be subdivided until a partition of 512 KiB for hosting the operating system. The other partition could be allocated to a process. Therefore the maximum size of a process is 256 MiB.*
4. *The size of the process depends on the logical address, so that the maximum size is 2^{32} bytes.*

Deliverable 6.2

Consider a paged virtual memory system with 32 bits logical addresses and a physical memory of 1 GiB. The page size is 8 KiB and page table and frame table entry sizes are 64 bits. Calculate:

1. The maximum size of a process in this system.
2. The size of the page table.
3. The size of the frame table.
4. When we have 10 active processes in the system, how many page tables are there in the system? And how many frame tables?

Deliverable 6.2 Solution

Physical memory = 1 GiB, Page size = 8 KiB

1. Process max. size = Logical Address Space
   Process max. size = 2^{32} B = 4 GiB
2. Page Table Size = Max. number pages of process $\times$ Entry Size PT
   Max. number pages of process = Process max. size / Page Size
   Max. number pages of process = 2^{32} B / 2^{13} B = 2^{19} pages
   PT Size = $2^{19} \times 2^{3}$ B = 2^{22} B
3. Frame Table Size = Number of physical memory frames $\times$ Entry FT Size
   Number of physical memory frames = Physical Memory / Frame Size
   Number of physical memory frames = 2^{30} B / 2^{13} B = 2^{17} frames
   Frame Table Sizes = $2^{17} \times 2^{3}$ B = 2^{20} B = 1 MiB
4. 10 frame tables, one per process. Only one frame table.

Deliverable 6.3 Exercise 1

Bit maps

1. Consider a system with 8GiB of main memory where the size of the asignation unit is 8 KiB. The free-space management is made through a bit map. Calculate the size of the bit map.

Deliverable 6.3 Exercise 1 Solution

Bit maps

1. Consider a system with 8GiB of main memory and the size of the asignation unit is 8 KiB. The free-space management is made through a bit map. Calculate the size of the bit map.
   No. bits in the bit map = Main memory size / Asignation unit size
   No. bits in the bit map = 8 GiB / 8 KiB =
   $2^3 \times 2^{30}$ B / $2^3 \times 2^{10}$ B = 2^{20}
   Bit map size = 2^{20} bits / 2^3 bits/B = 2^{17} B = 128 KiB

Deliverable 6.3 Exercise 2

Paged virtual memory

2. A virtual memory system has the followings characteristics:
   - Logical addresses: 64 bits
   - Physical addresses: 30 bits
   - Page size: 4 KiB
   - Entry page-table size: 32 bits
   - Entry frame-table size: 32 bits

   Answer these questions:
   1. Physical memory size.
   2. Size of the logical address space.
   3. Frame table size.
   4. Page table size.

Deliverable 6.3 Exercise 2 Solution

A paged virtual memory system:

- Logical addresses: 64 bits
- Physical addresses: 30 bits
- Page size: 4 KiB
- Entry page-table size: 32 bits
- Entry frame-table size: 32 bits

1. Physical memory size.
   Physical address = 30 bits $\longrightarrow$ Physical memory = 2^{30} B = 1 GiB
2. Size of the logical address space.
   Logical address = 64 bits $\longrightarrow$ Logical address space = 2^{64} B
3. Frame table size.
   Frame Table size = No. frames $\times$ Entry FT size
   No. Frames = Physical memory / Frame size
   Frame Table size = (2^{30} B / 4 KiB) $\times$ 4 B = (2^{30} B / 2^{12} B) $\times$ 2^{2} B = 2^{20} B
4. Page table size.
   Page table size = No. pages $\times$ Entry PT size
   No. pages = Virtual address space / Page size
   No. pages = 2^{64} B / 4 KiB = 2^{64} B/ 2^{12} B= 2^{52}
   Page table size = 2^{52} $\times$ 2^{2} B = 2^{54} B

Deliverable 6.3 Exercise 3

Format of memory addresses

3. Consider a paged virtual memory system with 32 bits logical address and a physical memory of 1 MiB. The page size is 2 KiB. Calculate:
   1. Logical address format, expressing the size (in bits) of each element.
   2. Physical address format, expressing the size (in bits) of each element.
4. Consider a segmented virtual memory system with 32 bits logical addresses and where the maximum size of segment is 16 KiB. Calculate the logical address format, expressing the size (in bits) of each element.

Deliverable 6.3 Exercise 3 - Solution

Logical Address Size = 32 bits, Physical Memory = 1 MiB, Page Size = 2KiB

1. Logical Address Format: | p | d |

   If Page Size = 2 KiB = 2^{11} B $\longrightarrow$ d = 11 bits

   If Logical Adress Size = p + d = 32 bits $\longrightarrow$ p = 21 bits

2. Physical Address Format: | m | d (11 bits) |

   If Physical Memory = 1 MiB = 2^{20} B $\longrightarrow$ Physical Address Size = 20 bits

   If Physical Address Size = m + d = 20 bits $\longrightarrow$ m = 9 bits

Logical Address Size = 32 bits, Maximum Segment Size = 8 KiB

Logical Address Format: | s | d |

If Maximum Segment Size = 16 KiB = 2^{14} B $\longrightarrow$ d = 14 bits

If Logical Address = s + d = 32 bits $\longrightarrow$ s = 18 bits

Deliverable 6.4

Consider a paged virtual memory system where:

- Logical address size: 32 bits
- Page size: 4 KiB
- Page Table entry size: 32 bits

The system uses two-level page tables. Determine the format of logical addresses specifing the size (in bits) of each component.

Deliverable 6.4 - Solution

Logical Address Format: | p1 | p2 | d |

Logical Address Size = p1 + p2 + d

Page Size = 4 KiB = 2^{12} B $\longrightarrow$ Offset = d = 12 bits

Entries in a Page Table = Page Size / Entry Size

Entries in a Page Table = 2^{12} B / 2^{2} B = 2^{10} entries $\longrightarrow$ We need 10 bits for p1 and 10 bits for p2.

Logical Address Format: | 10 | 10 | 12 |

Deliverable 6.5 Exercise 1

Multilevel page tables

1. Consider a virtual memory system which uses two-level page tables:
   - Logical address size: 32 bits
   - Page size: 8 KiB
   - Page table entry size: 32 bits

   Calculate logical address format considering second level page tables don't show internal fragmentation.

Deliverable 6.5 Exercise 1 - Solution

Multilevel page tables

1. Consider a virtual memory system which uses two-level page tables:
   - Logical address size: 32 bits
   - Page size: 8 KiB
   - Page table entry size: 32 bits

   Calculate logical address format considering second level page tables don't show internal fragmentation.

   - Page size = 8 KiB = 2^{13} B, then **d=13 bits**
   - p1 + p2 + d = 32 $\longrightarrow$ p1 + p2 = 32 - 13 = 19 bits
   - No. entries in a PT = Page size / Entry PT size = 2^{13} B /4 B = 2^{11} entries
   - p1 $\leq$ 11 bits and p2 $\leq$ 11 bits
   - If second level page tables don't show internal fragmentation, then the component **p2 = 11 bits**
   - Therefore p1 = 19 - p2 $\longrightarrow$ **p1 = 8 bits**

Deliverable 6.5 Exercise 2

Multilevel page tables

2. Consider a paged virtual memory system which uses three-level page tables:
   - Logical addresses: 32 bits
   - Physical addresses: 24 bits
   - Page size: 2 KiB
   - Page table entry size: 32 bits

   Calculate:
   1. Number of bits for offset.
   2. Maximum number of entries in a page table.
   3. Maximum number of bits for each pagination component.
   4. Logical address format considering first and second level page tables don't show internal fragmentation.
   5. Internal fragmentation in this system.

Deliverable 6.5 Exercise 2 - Solution

Logical address = 32 bits, Physical address = 24 bits, Page size = 2 KiB, PT Entry = 32 bits, three-level PT

1. Page size = 2 KiB = 2^{11} B $\longrightarrow$ d = 11 bits
2. The size of all page tables is one page, then No. entries in a PT = Page Size / PT entry size = 2 KiB / 32 bits = 2^{11} B / 4 B = 2^9 entries
3. To address 2^9 entries we need 9 bits.
4. If first and second level page tables don't show internal fragmentation, $\longrightarrow$ the components p1 = p2 = 9 bits. The logical address is:

p1 (9 bits)	p2 (9 bits)	p3	d (11 bits)

p3 = (32 - p1 - p2 - d) bits = 3 bits

Deliverable 6.5 Exercise 2 - Solution

Logical address = 32 bits, Physical address = 24 bits, Page size = 2 KiB, PT Entry = 32 bits, three-level PT

5. Internal Fragmentation = Intern. Frag. 1-level PT + Intern. Frag. 2-level PT + Intern. Frag. 3-level PT
   Internal Frag. N-level = No. N-level PT $\times$ Internal Frag. N-level PT
   Internal Frag. N-level PT = Page Size - $2^{pN} \times$ PT entry size
   No. N-level PT = $2^{p1} \times 2^{p2} \times \cdots \times 2^{p(N-1)}$
   Therefore, in this system:
   Intern. Frag. 1-level PT = Intern. Frag. 2-level PT = 0
   No. 3-level PT = $2^9 \times 2^9$ = 2^{18} PT
   Int. Frag. 3-level PT = 2^{11} B - (2^3 entries $\times$ 4 B/entry) = 2^{11} B - 2^5 B = 2016 B
   Total Fragmentation = 2^{18} TP $\times$ 2016 B = 528.482.304 B

Deliverable 6.5 Exercise 3

Inverted page table

3. Consider a paged virtual memory system which uses inverted page table:
   - Logical addresses = 32 bits
   - Physical addresses = 24 bits
   - Page size = 2 KiB
   - Page table entry size = 32 bits

   1. Calculate the size of inverted page table.
   2. How many inverted page tables are there?

Deliverable 6.5 Exercise 3 - Solution

Logical address = 32 bits, Physical address = 24 bits, Page size= 2 KiB, PT entry size = 32 bits, inverted PT

1. Physical address format $\boxed{\text{m}}\boxed{\text{d (11 bits)}}$
   24 bits = m + d
   m = (24 - 11) bits = 13 bits
   Inverted PT Size = 2^m entries $\times$ PT Entry Size
   Inverted PT Size = 2^{13} entries $\times$ 4 B/entry = $2^{15}B$ = 32 KiB
   Another way to obtain the solution:
   Physical address = 24 bits $\Longrightarrow$ Physical Memory = 2^{24} B
   No. frames = Physical Memory / Frame size
   No. frames = 2^{24} B / 2 KiB = 2^{24} B / 2^{11} B = 2^{13} frames
   Inverted PT size = No. frames $\times$ TP entry size
   Inverted PT size = 2^{13} frames $\times$ 2^2 B = $2^{15}B$ = 32 KiB
2. Only one inverted page table is needed in the system.