Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informática de Sistemas y Computadoras (DISCA)
Universitat Politècnica de València

Part 3: Memory management

Seminar 10
Contiguous and sparse memory allocation exercises

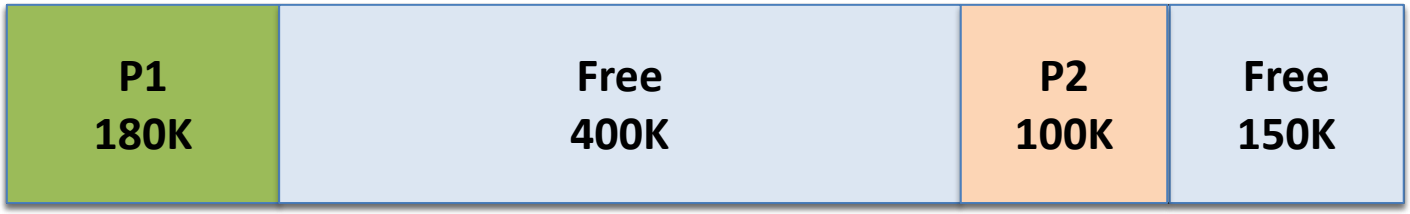
fSO

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA

Exercise 1.1

In a system which memory is managed using **variable size partitions with compaction.** At a certain moment there is the following memory occupation:



The allocation policy used is **best fit**. In the system input queue there are three jobs in the order: **P4(120K), P5(200K) and P6(80K)**, that have to be selected in FIFO order. Suppose that none of the active processed finishes, then after trying to put in memory all jobs in the input queue:

a) ¿How many partitions remain free?¿What are their sizes?

b) If compaction is done in this situation, ¿what processes should be moved in such a way that the number of moving bytes be the least and only one hole will remain?

c) If the base registers of every process are, respectively B1, B2, B3, B4, B5 and B6, ¿how have changed the base registers corresponding to the processes moved by compaction?

Exercise 1.2

Consider a system with multiple partitions with hardware support based on base and limit register technique.

Given a program P that occupies T words and is allocated in physical memory from address C:

a) ¿What is the value for every program P register?

b) ¿What is the logical address range that P can access?

c) ¿What is the physical address range where addresses accessed by P are allocated?

Exercise 1.3

A computer has a 32 bit address processor and 128 Mbyte physical memory. Its OS occupies 64Mbyte (allocated in lower memory addresses) and it uses a variable partition memory manager. In this system user processes have a minimum size of 16Kbyte.

a) ¿What is the maximum size of a user process that can be executed?

b) ¿What is the maximum multiprogramming level allowed?

c) Describe a situation with 2 processes (Pa, Pb) allocated in memory and compaction must be applied to allocate a new 20Mbyte process Pc.

Content

- Exercise 1: Contiguous allocation
  - Exercise 1.1: Variable partitions
  - Exercise 1.2: Base and limit register
  - Exercise 1.3: Multiprogramming level
- **Exercise 2: Sparse allocation**
  - Exercise 2.1: Logical and physical address formats
  - Exercise 2.2: From logical to physical addresses
  - Exercise 2.3: From logical to physical addresses
  - Exercise 2.4: From physical to logical addresses
  - Exercise 2.5: From physical to logical addresses
- Exercise 3: Table size and fragmentation
  - Exercise 3.1: Table size
  - Exercise 3.2: Internal fragmentation

Ejercicio 2.1

- In a system with 1GByte of main memory and 4GByte of logical addressing capability a 2 level paging (same number of addressing bits) memory management system is going to be implemented. The page size is 16KBytes and in every page table has 8 byte page descriptors.

  What are the logical address and the physical address formats? Give their fields and corresponding number of bits for every field

Exercise 2.2

A CPU generates the logical addresses**: 612, 38 y (3,62)**. Describe the physical addresses after translating them applying the proposed memory management policies. If the logical address cannot be generated or translated tell **ERROR.**

a) **Variable partitions** with base and limit registers

Base register	Limit register
150	220

b) **Paging** with 128 page size, the page table is:

Page table

0	1
1	4
2	2
3	5

c) **Segmentation**, the segment table is:

Segment table

	Base	Límite
0	200	20
1	50	10
2	105	49
3	320	70

Exercise 2.3

The following situations correspond to one or more processes generating logical addresses. Obtain the corresponding physical addresses given by every memory management policy. In case of problem tell **ERROR.**

a) Variable partition. The logical addresses generated are: process B 530, process A (0,130), process C 1046. The base and limit registers values are:

Process	Base	Limit
A	0	1360
B	4020	6300
C	1400	2600

b) Paging with 256 page size. Suppose that logical addresses are generated by only one process: 530, (0, 130) and 1046. Process table page content is:

Page table

0	4
1	5
2	3
3	6

c) Segmentation with 4 segments. The following logical addresses are generated by a single process: **530, (0, 130), (1,25) y (4,50).** The content of the segment table is:

Segment table

	Base	Limit
0	200	20
1	50	50
2	105	49
3	320	70

Exercise 2.4

In a system with 64Kbyte of physical memory, there is an access to physical address **27214** as a logical address translation generated by process P1. P1 size is 15535 byte. Obtain the logical addresses generated by P1 in order to happen the previous access, considering:

a) Paging with 4K byte page size.

b) Segmentation with 16K byte segment size. Consider that segments are always allocated in addresses multiple of 16K and that P1 fits inside a single segment.

Exercise 2.5

A memory manager based on **two level paging**, with 1K page size. The logical address space is 4Mbyte and the first level page table has 1024 entries.

Obtain what possible logical address(es) can correspond to physical address 2516.

Content

- Exercise 1: Contiguous allocation
  - Exercise 1.1: Variable partitions
  - Exercise 1.2: Base and limit register
  - Exercise 1.3: Multiprogramming level
- Exercise 2: Sparse allocation
  - Exercise 2.1: Logical and physical address formats
  - Exercise 2.2: From logical to physical addresses
  - Exercise 2.3: From logical to physical addresses
  - Exercise 2.4: From physical to logical addresses
  - Exercise 2.5: From physical to logical addresses
- **Exercise 3: Table size and fragmentation**
  - Exercise 3.1: Table size
  - Exercise 3.2: Internal fragmentation

Exercise 3.1

In a **2 level paging** memory system, the **logical address spaces are 8 Gbyte** with **2 Kbyte page size**, and the **first level table has 256 entries**. Obtain the size of the second level page tables considering 4-byte page descriptors.

Exercise 3.2

A memory manager uses **2 level paging** with 16 bit logical addresses, 256-byte page size and second level page table with 16 entries.

Obtain:

a) **Maximum process size** in bytes.

b) **P0, P1 and offset** for the following logical addresses: 0x9134, 0x5634, 0x4500, 0x0012.

c) **Total internal fragmentation** of a set of processes with the following sizes: P0 2688 Bytes, P1 1984 Bytes, P2 1344 Bytes and P3 3264 Bytes.