Operating Systems
Memory Management – Solution

1. Describe two differences between logical and physical addresses.

   Answer -
   A logical address does not refer to an actual existing address; rather, it refers to an abstract address in an abstract address space. Contrast this with a physical address that refers to an actual physical address in memory. A logical address is generated by the CPU and is translated in to a physical address by the memory management unit (MMU) (when address binding occurs at execution time). Therefore, physical addresses are generated by the MMU.

2. Why are page/frame sizes always powers of 2?

   Answer -
   Recall that paging is implemented by breaking up an address into a page and offset number. It is most efficient to break the address into X page bits and Y offset bits, the offset represents the position of an octet from the start of the page. Since both the page size and frame size are the same, translating a logical address to a physical address only requires the replacement of the page number by the frame number (a concatenation operation). 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.

3. Consider a logical address space of eight pages of 1024 bytes each, mapped onto a physical memory of 32 frames.

How many bits are there in the logical address? Identify in the logical address the bits used as an offset and the bits used as the page number.
Answer -
Logical address: 13 bits (8 pages X 1024 bytes/page = 8196 bytes – requires 2^{13} addresses)
Bits for offset: Bits 0 to 9 (10 bits for offset into 1024, 2^{10} byte page), 3 bits identify page number (2^3= 8 pages)

How many bits are there in the physical address? Identify in the physical address the bits used as an offset and the bits used as the frame number.
Answer -
Physical address: 15 bits (32 frames X 1024 byes/frame = 32768 bytes, requires 2^{15} addresses).
Bits for offset: Bits 0 to 9 (10 bits for offset into 1024, 2^{10} byte frame), 5 bits identify frame number (2^5= 32 frames)

Partition Allocation Methods

In Partition Allocation, when there are more than one partition freely available to accommodate a process's request, a partition must be selected. To choose a particular partition, a partition allocation method is needed. A partition allocation method is considered better if it avoids internal fragmentation.

Below are the various partition allocation schemes :

1. First Fit:

In the first fit, partition is allocated which is first sufficient from the top of Main Memory.

2. Best Fit:

Allocate the process to the partition which is first smallest sufficient partition among the free available partition.

3. Worst Fit:

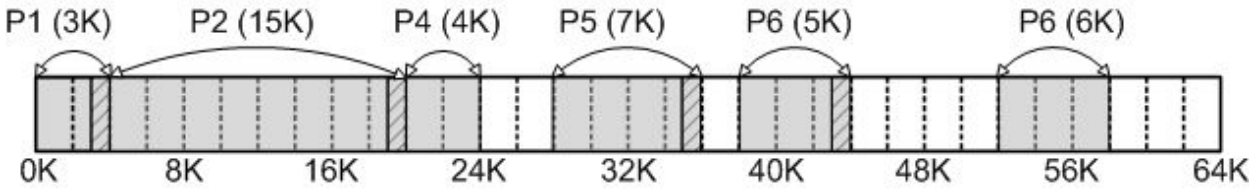
Allocate the process to the partition which is largest sufficient among the freely available partitions available in the main memory.

4. Next Fit:

Next fit is similar to the first fit but it will search for the first sufficient partition from the last allocation point.

==========

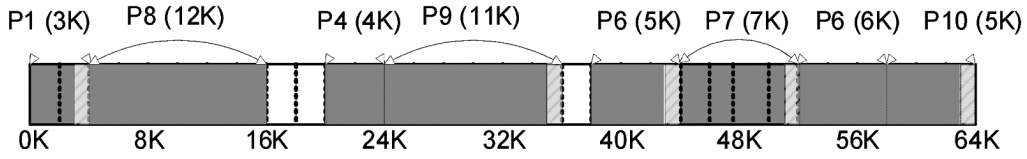
4. Consider a computer using dynamic contiguous allocation, where each allocated block is a multiple of 2kb. Consider the following situation:



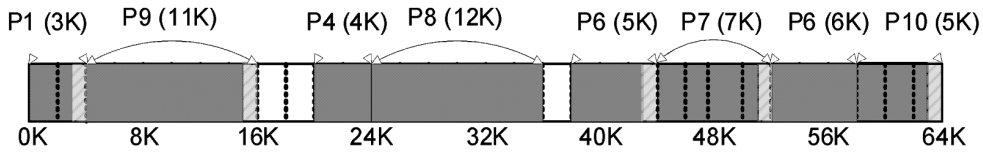
Consider the following sequence of allocating/deallocating

- Allocate process P7, size 7kb
- Deallocate process P2
- Deallocate process P5
- Allocate process P8, size 12 kb
- Allocate process P9, size 11 kb
- Allocate process P10, size 5k

a) Draw pictures describing the final allocation using first-fit and best-fit allocation algorithms.



First fit



Best fit

b) What is the internal fragmentation at the end? What about external fragmentation?

Internal Fragmentation: 5K (5 processes have 1 k unused)
External: 6K – Three holes of 2k unused by processes

==========

Examples:

1. Given five memory partitions of 100Kb, 500Kb, 200Kb, 300Kb, 600Kb (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of 212 Kb, 417 Kb, 112 Kb, and 426 Kb (in order)?
Which algorithm makes the most efficient use of memory?

Solution:
First-fit:
212K is put in 500K partition
417K is put in 600K partition
112K is put in 288K partition (new partition 288K = 500K - 212K)
426K must wait

Best-fit:
212K is put in 300K partition
417K is put in 500K partition
112K is put in 200K partition
426K is put in 600K partition

Worst-fit:
212K is put in 600K partition
417K is put in 500K partition
112K is put in 388K partition
426K must wait

In this example, best-fit turns out to be the best.

==========

Examples 2:

2. Assuming a 1 KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):
a. 2375
b. 19366
c. 30000
d. 256
e. 16385

Solution:
a. page = 1; offset = 327
b. page = 18; offset = 934
c. page = 29; offset = 304
d. page = 0; offset = 256
e. page = 16; offset = 1

==========

3. 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?
a. 0,430
b. 1,10
c. 2,500
d. 3,400
e. 4,112

Solution:
a. 219 + 430 = 649
b. 2300 + 10 = 2310
c. illegal reference, trap to operating system
d. 1327 + 400 = 1727
e. illegal reference, trap to operating system