IT2164/IT2561 Operating Systems

# Tutorial 8

**Memory Management**

Attempt the following questions before you attend tutorial.

1. What are the 3 ways to assign addresses to the instructions in a computer program? Suppose we have a computer system in which we require to shift the program in memory whenever other programs get terminated or new programs come in. Suggest which method of address binding we can use and why.

* The 3 methods are compile-time binding, load-time binding and run-time binding
* The requirement that programs be able to be moved when they are already in memory requires the use of run-time binding. This is because by calculating the actual address when the instruction is executed, programs are able to be moved anywhere in memory during execution. The other 2 methods do not allow this as they assign fixed addresses during compile-time and loading-time

1. Under what circumstances do external fragmentation and internal fragmentation occur? Can they occur together?

* External fragmentation occur when there is dynamic allocation of memory in blocks of different sizes (variable partition). Over time, many small separate blocks will be created due to processes being added and terminated
* Internal fragmentation will occur when memory is divided into fixed-size blocks (fixed partition) and allocated in blocks (note that the sizes of each block may or may not be the same)
* No, external and internal fragmentation cannot occur together in a system. They are mutually exclusive

1. What is compaction? What problem does it solve? Which address-binding method do you think a feature like compaction requires?

* Compaction is the relocation of processes in memory in order to recombine fragmented free memory
* Compaction solves the problem of external fragmentation, which is the result of variable memory partitioning
* Compaction requires run-time binding, since processes are likely to be moved when compaction is done

1. The following diagram shows a snapshot of the memory at time T0.

Program 1

–

100

K

Program 2

–

250

K

Program 3

–

K

150

Empty

–

160

K

Empty

–

K

140

Empty

–

200

K

1000

K

K

0

* 1. At time T1, the system undergoes compaction. With the aid of a diagram, show the result after compaction.
* Program 1 – 100K
* Program 2 – 250K
* Program 3 – 150K
* Empty – 500K
  1. What is the value of the relocation register for Program 2 after it is moved during compaction?

During compaction, Program 2 will be moved up by 160K. Since lower addresses

are at the top, all addresses will be reduces, hence the relocation register value will

be a negative value

-160K = -160 \* 1024 = -163840

* 1. Before compaction, the memory address of the instruction ‘Load =7, R1’ is 368640. What is the new memory address after compaction?

The new address of the instruction “Load = 7. R1” originally at 368640 will be

reduced by 163840.

Hence the new memory address is (368140-163840) = 204800.