

SINGAPORE POLYTECHNIC
SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING
ET0023 OPERATING SYSTEMS

TUTORIAL 7 – Deadlocks & Memory Management

1. A set of processes is in a deadlock state when every process in the set is waiting for an event that can be caused only by another process in the set.

Notes:

- a) occurs with two or more processes
- b) events are usually resource acquisition or release.

Deadlocks only occur with non-preemptive resources because if the resource is removed from the process, the process will fail. Preemptable resources can be removed from a process without any ill effects.

2.

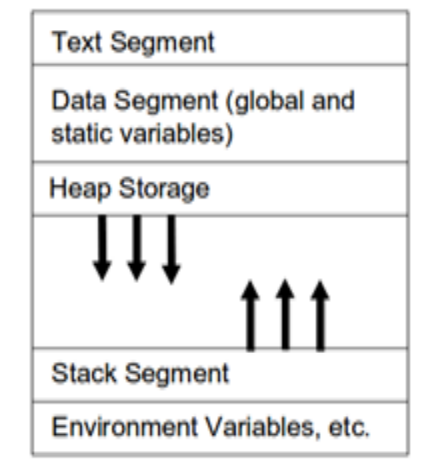
Deadlock Conditions	Possible solution techniques
Mutual exclusion condition Each resource assigned to one process or is available	Spool all resources
Hold and Wait condition Process holding resources can request additional resources	Processes must request all resources initially before running
No pre-emption condition Previously granted resources cannot be forcibly taken away.	Forcibly terminate/take away the resource
Circular wait condition A chain of two or more processes each waiting for the resource held by the next member of the chain.	Order resources numerically to ensure no circular waits occur.

Modern operating systems do implement some of the above techniques, however, must use the "Ostrich" solution and pretend that there is no problem.

Reasons: deadlocks are very rare, cost of fixing solution is very expensive.

3. Read: http://en.wikipedia.org/wiki/Memory_management_unit
 The prime objectives of the MMU is to efficiently allocate memory to processes, utilizing as much of the memory available and as little fragmentation as possible.

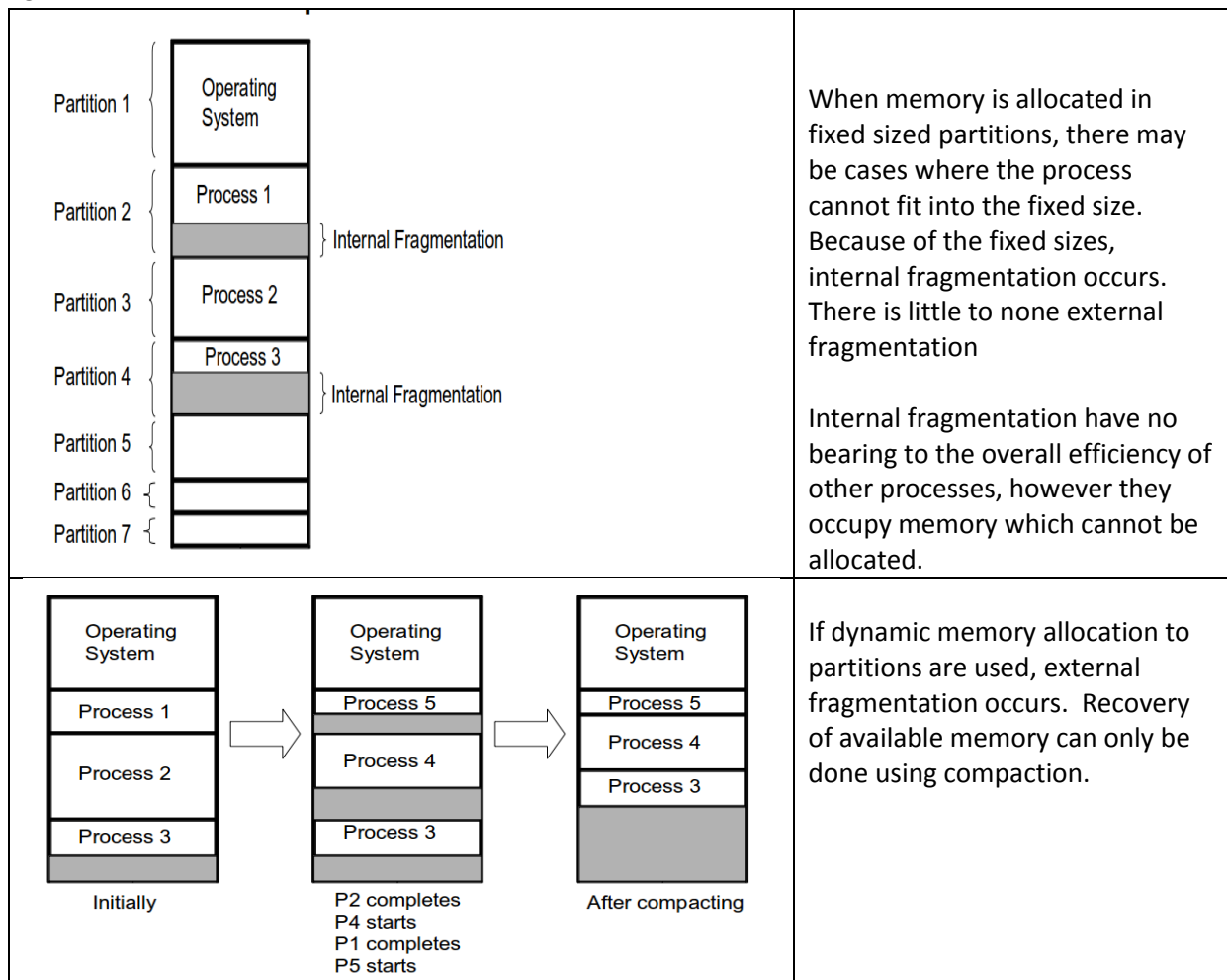
4. Allocation of memory in a process:



Text segment holds string constants
 Data segment holds global and static variables
 Heap storage is used for dynamic memory used in the program. Heap storage increases towards a limit.
 Stack segment is used to hold stack variables and stack area grows towards lower memory.
 Environment variables are set in an area that is accessible to both the OS and process. The size of the memory allocation is fixed.

MMU allocates more memory than required, so that it caters for dynamic memory/variables

5.



6. See lecture 7 slides 32-36, Students have to be familiar with the concept of pages and frames.