

# Operating System Concepts Weekly

## Week 6

Lucas van der Laan  
s1047485

### 1

- a) Internal and External fragmentation.  
**Internal fragmentation:** Occurs in paging and fixed partitioning.  
**External fragmentation:** Occurs in segmentation and dynamic partitioning.
- b) Paging avoids the need for external fragmentation and we expect internal fragmentation when the allocated memory size is not a multiple of the page size. The modulo of the allocated space and the page size is the amount of internal fragmentation we have.
- c) Both problems are related to internal fragmentation, in the first case, the padding between the fields is internal fragmentation and for the second issue, this is only an issue with internal fragmentation, because with external fragmentation we can fragment the memory up into multiple places and no longer, so it is not difficult to reuse that memory.

### 2

- a) Page size:  $2^{24}$   
Address Space:  $2^{64}$   
Page table entry size:  $2^{64}$  bits, also known as 8 bytes  
Address Space / Page Size:  $2^{64}/2^{24} = 2^{32}$  pages.  
Pages \* table entry size:  $2^{32} * 8 = 34.359.738.368$  bits, which is 4 GiB.

### 3

- a) Because the threads run in the same process and oftentimes will have to access the same data. This is not a problem, because generally speaking all threads of a program work in tandem, unlike multiple processes which can have different reasons of existing.
- b) Yes, but only when in kernel mode, this is because we do not want any other program to modify an existing process, as it can be done by malicious code. So to ensure that this can be done by non-malicious code, we run it as root/administrator, so we can modify other processes when needed. The reason why we should allow it? Simply to give the user and their programs the capability of interacting with one another, so you can read/write from/to any other process.