Module: CT213 Computer Systems and Organisation (Answer Sheet)

Date: 22/01/2021
Student name: Daniel Hannon
Student ID: 19484286

Student statement of academic integrity: In submitting this work I confirm that it is entirely my own. I acknowledge that I may be invited to online interview if there is any concern in relation to the integrity of my exam.

- We suggest to immediately (before starting to type answers) **save this file to a known location and filename**.
- Please answer questions under the question numbers below, e.g. Question 1(a).
- Remember to regularly save your work.
- If you need to create diagrams, you can do so on paper and then add photographs to this document.
- Text answers can be typed out **or** written on paper and then added as an image.
- When you have completed the exam, **save this file as a Single PDF** and submit it on Blackboard.
- If you run into submission difficulties on Blackboard, you should immediately (within the designated time) send your exam as an email attachment to ali.hasnain@nuigalway.ie Only do this if there are problems with Blackboard. If there are no problems, you **only** need to submit through Blackboard.

Question 1 (a)

Operating System Classification criteria consists of the following:

Process Scheduling: This is the means in which the execution of a program is scheduled so several programs can run in a pseudo-simultaneous fashion without affecting the data that is required by another program

Memory Management: this is the means in which the Use of RAM is coordinated by the operating system to allow multiple programs to have data stored in memory without affecting the integrity of data from another.

I/O Management: This is the means in which an Operating System deals with communications between I/O devices and the computer itself, it can be interupt driven or run on a basis which requires Polling of I/O Devices., the latter is usually picked.

File Management: This is the means in which an operating system coordinates data stored in your Hard Disk, it pertains to the retention of records pertaining to the location of files within your system so they cannot be lost or overwritten by accident.

Question 1 (b)

Batch: A Batch operating system coordinates processes on a FIFO(First In, First Out) basis, this makes processes easier to coordinate as they are guaranteed to finish in the same order in which they go in

Memory Management: Memory is divided into System Memory and Program memory

I/O Management: A job has specific access to I/O Devices

Time Sharing:
A Time Sharing operating system utilizes an algorithm such as Round Robin to coordinate the execution of programs and as a result, allows multiple programs to be present within the machine at the same time.
In regards to memory management, it has protections to ensure the processes can run without affecting the result of another.
There exists protections for I/O and File Management too.

Real Time Operating System: These are time critical systems so tasks are scheduled in order of priority rather than the order in which they are

loaded into the machine. I/O is interrupt driven, they do not necessecarily have any form of File Management.

Distributed Operating System: These are run across multiple servers and could be implemented any way of the three above. They have a high priority for security.

Question 2

Process Manager: The Process manager is responsible for the coordination of the execution of processes, it schedules time slices for the processes to avail of the CPU, checks for deadlock between processes and resolves it, it also provides a translator between virtual memory addresses that the processes use and the actual memory address which is accessed from the memory manager. As a result of this, it prevents processes from illegally accessing memory that has been designated to another process. It is also responsible for changing what's contained within the various registers of the CPU during context switches.

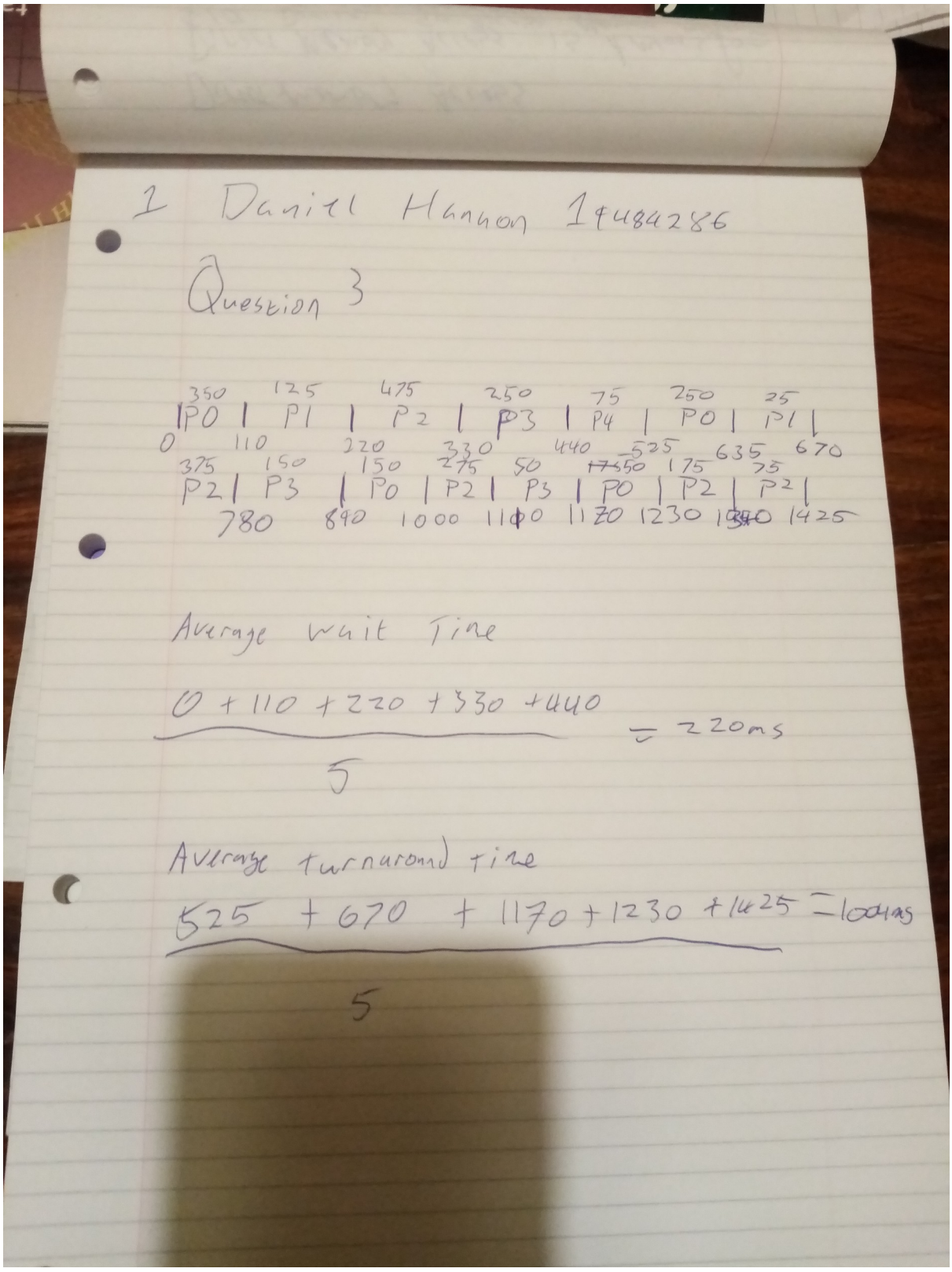
File Manager: The file manager is responsible for the indexing of files within the storage media on the computer, It controls any read/write operations as a result it can load a file into memory or send it to a device (I.e A printer)

Memory Manager: the memory manager keeps track of the areas in RAM that are currently in use, it is also responsible for indexing and allocating space in memory for processes that are being run or files that are loaded. It can perform both Read and Write operations within RAM and as RAM is volatile, it also contains a clock to ensure that the memory is refreshed regularly to prevent any data loss.

Device Manager: The device manager keeps track of any and all I/O devices that are connected to the computer, it is responsible for any Data sent to or from these devices, as not all devices have the same means of communication, it avails of Drivers to translate device instructions to instructions that are understood by the computer and vice versa

---

Question 3



Average Wait time : (0 + 110 + 220 + 330 +440)/5 = 220ms

Average Turnaround time : (525 + 670 +1170 + 1230 +1425)/5 = 1004ms

Question 4

Critical Section: A critical section of a program is a section in which it must run to completion without interruption to ensure the integrity of the data: eg. Arithmetic Operations

Deadlock: Deadlock is when two or more processes are waiting for the other to exit their critical sections before they can use the resource presented, as none of them are in their critical section, the resource is never used and none of the programs run to completion

Interrupts: An Interrupt is a signal sent by a higher priority process to take control of a resource, as a result they eliminate the need for polling: I.E: A computer with no Direct Memory Access for keyboard inputs would have an I/O interrupt sent to the CPU every time that a key is toggled.

Context Switch: Context switching is the process of changing the process that has control of the CPU in any program schedulling algorithm, this largely consists of moving the content of the CPU to a previously specified location within memory and inserting the content from another specified location within memory.

Mutual Exclusion: Mutual Exclusion is a means in which we can prevent two or more processes from being in their critical section at the same time, this can be done by disabling interrupts while a process is in a critical section or by implementing something like semaphores.

Pre-emptive scheduling: Processes are scheduled to run in the order of priority this also allows higher priority processes to force lower priority tasks to yield resources so that they can use them instead.
This can make some processes never run if they are of a low priority and many high priority tasks are scheduled ahead of them.

Process Messages: Process messages are messages sent or recieved to the CPU they can be instructions to Yield, Sending a read/write instruction to an I/O Bus, among many other things.

Starvation: Starvation is achieved when multiple processes require the use of a specific resource in order to run to completion and multiple processes of a high priority repeatedly successively take control of the resource ultimately potentially preventing a low-priority process of ever taking control of the given resource and in turn, never running to completion.

Question 5 (a)

1 + (2 | 9) is 2 9 | 1 + in postfix

PUSH 2
PUSH 9
PUSH 1
POP 2
POP 9
PUSH 2|9=11
  0 0 1 0 = 2
| 1 0 0 1 = 9
  1 0 1 1 = 11

Follows Truth Table
 | 0 1
 0 0 1
 1 1 1

POP 11
POP 1
PUSH 11+1 = 12

Question 5 (b)

Machine Language: Machine Language code is binary instruction, this is not easily readable by humans and it is CPU-Exclusive

Assembly: Assembly is an instruction set that is human readable and is Architecture Specific, this allows people to write code for specific machines, Assembly is a compiled language which means you must run it through a compiler which in turn transforms it into machine code which is then run by the computer. Assembly is often made up of a small instruction set (E.G RISC/CISC) and is not too abstract from the processes that the machine performs.

High-Level Languages: These are far more abstract than assembly making them easily readable and writeable, they do not require as much of an in depth knowledge to perform tasks on a machine and they are often portable as they follow a predefined convention or “standard” in which interpreters/compilers for varying machines which adheres to this standard must exist to run the code written in them. A good example of a High Level Language would be Javascript or Python as they are extremely abstract.

Question 6 (a)

Record blocking is the process of storing files in blocks within the system so they can be quickly accessed by users/processes. This also makes it easier to determine how much space is occuppied within a system as they are filled in a linear fashion.

Question 6 (b)
Fixed Blocking is the means of allocating groups of records/files in blocks of a set size and the each record must fit in its entirety within a block. Each record is the same size

Variable length Spanned Blocking allows records to be stored across multiple blocks in a linear fashion. Records are not required to be the same length.

Variable Length Non-Spanned Blocking allows records of variable length to be stored within blocks of a fixed size, unlike the above they cannot span multiple blocks

Question 6 (c)
Fixed Blocking and Variable Length Non Spanned blocking are vunerable to Fragmentation of records as they can often leave unused data at the end of the block in which they are in charge of, in the case of fixed blocking this space can often never be occupied where with variable length if defragmentation is performed the unused areas may become populated.

Variable Length Spanned Blocking is nowhere near as susceptible to fragmentation of data as the end of a record that started in the previous block, is present in the next.

Question 9

Multiprogramming: Multiprogramming is when more than one process can be loaded into a computer at the same time, in terms of memory management this requires the Memory to be broken into multiple blocks that can store the data required by multiple programs, In terms of the operating system, The process itself is typically not designed to access an arbitrary memory address so it must provide Page Tables which contain the Actual Memory address of content required by a process rather than the virtual address that it accesses. In terms of device management, this results in the existence of DMA or Direct Media Access where an I/O Device has direct access to a certain location within memory to prevent Interrupts of processes that may be currently running, I.E monitor refreshing is performed asynchrously to the CPU.

Privileged Mode: Privileged mode exists so that only essential components of the operating system have direct access to the physical components of the hardware I.E actual memory addresses. This is done to prevent unnessecary damage being caused by malicious programs or by an incompetent user. The User has access to User Mode which allows them to run tasks without the fear of accidentally/unnessecarily inflicting damage on the system

Thread: A thread is a lightweight process that is owned by a parent process, it has access to the memory locations and resources in which it's parent has access to and nothing else. A Thread can only belong to one process where a process can have multiple processes. A Thread executes the program data of a process.

Protection: Memory protection such as segmentation is performed by the operating system so Several processes cannot access the same area of memory (Unless they are threads of a parent process) and in turn protects the integrity of the data stored in memory by a given process.

Kernel: The Kernel is the core of the operating system, this is where all the essential processes of an operating system exist, they are run in supervisor mode and they often have little to no abstraction in terms of access to system resources. The processes of anything executed at kernel level cannot be altered by something running in user mode.

Thrashing: Thrashing is caused when memory is excessively used by processes, this results in a large number of page faults and a potential loss of integrity of data that may be stored in memory.
