**Operating Systems Assignment**

Contents

[Question 1 - Complete 1](#_Toc531607403)

[Question 2 - Complete 2](#_Toc531607404)

[Question 3 - Incomplete 3](#_Toc531607405)

[Question 4 - Incomplete 3](#_Toc531607406)

[Question 5 - Complete 4](#_Toc531607407)

[Question 6 4](#_Toc531607408)

[Question 7 4](#_Toc531607409)

[Question 8 4](#_Toc531607410)

## Question 1 - Complete

1. User mode doesn’t have the ability for executing code to directly access hardware or reference memory. Kernel mode allows executing code to have unrestricted access to the hardware within the machine. Kernel mode is responsible for the most critical executions within the computer, while User mode is responsible for most executions within the use of a computer. Kernel mode processes that crash are more likely to stop the entire machine from working, while User mode crashes may stop a program or operation.
2. The three common events that lead to a process switch are: Clock interrupt, I/O interrupt, and Memory fault. Clock interrupt will cause a process switch when the time taken for the process to run has lapsed – its time slice expired. An I/O interrupt will occur when a process requires an I/O event and is taking up space waiting for it. Memory faults occur when a process requests memory from secondary memory. As another process is required to retrieve this data, the process that requested the data is swapped.
3. Threads are smaller than processes. Theoretically, a process can be made of threads while a thread cannot be made of processes. Threads within the same process share the same address space, allowing threads to read and write to the same data locations – while the same is possible for processes using IPC, it is difficult and resource intensive. Multi-threading allows for faster processing of a process, while multi-tasking allows the running of multiple processes – however these tasks do not run concurrently and thus are not faster because of multi-tasking.
4. Semaphores are a way of sending signals between processes, preventing the processes from trying to run at the same time. General semaphores allow a numerical value to be assigned and altered, while a Binary semaphore only uses 0 and 1. Both types of semaphores can be accessed by using the wait() and signal() operations regardless of type.
5. Deadlocks occur when a process or processes are waiting for at least one more to finish using a resource, while that one also waits for a process to finish using a resource. Livelock prevents a thread from continuing because it is too busy to make progress. This can often occur if a thread is used to respond to another thread. Starvation occurs when a process is unable to use the required resources because they are being used by another process for a long period of time. Deadlock is the most common issue of the three, and the easiest to resolve. Starvation and Livelock are harder to deal with as they occur without being completely stuck but have a possibility to resolve themselves if the dependencies resolve themselves.
6. Dynamic Partitioning allows logical processors, memory, and adapters to be dynamically added and removed from logical partitions. This is good for when the computer requires different quantities of storage space during runtime as it allows many different processes to have the right amount of storage allocated. An issue caused by this is that as more processes start and end the memory locations become more and more fragmented due to the allocated memory being given an address based on what was available at the time. This is avoided by using four methods of partitioning methods: Best fit, Worst fit, First fit, and Last fit. These allocate memory in different ways to reduce the amount of wasted space within the memory. Each of these methods are used during memory allocation to attempt to fill in holes in the memory usage to make more efficient use of the resources available.
7. Multi-Level Page tables can be thought of as pages of pages. The pages point to addresses within other pages rather than to a memory address. Another way to describe it is a hierarchical file structure within a computer, where a folder can hold folders of files that are semi-related. One issue with this is that a program can be slowed down by several times its usual running speed. Inverted Page Tables use one entry per physical page within a linear array, where each physical page is mapped to a virtual page. Due to this, each entry is mapped to a virtual page rather than a physical page. This differs from a regular page table as the address that is stored doesn’t exist. A regular page table can record many entries within one table. Inverted page tables can only store one. This means that to record data for many processes it would require many page tables.
8. Starvation can be caused by simplistic scheduling algorithms if a process begins to run but the process requires data that is being used by another process which blocks its ability to run. If a priority system is in place it can cause starvation as, if a high priority process never releases a resource, a lower resource may be unable to process due to a lack of access to the resource. One common approach to prevent this is to allocate each process a set time to process, after which it is paused, and another process runs. This prevents long periods of a process from being able to run as it periodically releases the resources held by processes and allows another process the opportunity to run.
9. Direct Memory Access (DMA) allows data to pass from an attached device straight to the memory on a computer’s motherboard. This is possible due to the configuration of some bus architectures. This allows the processor to perform other tasks while a task (e.g. I/O operations) to take place. This frees up the CPU to perform other more critical tasks.
10. Indexed sequential files are faster than sequential due to the index. Data is stored with an index generated by the data itself. If this process is known, the sought data can be run through this to find the index that the data is likely being held in.

## 

## Question 2 - Complete

1. This is a safe state as P4 can be allocated more resources to get all the resources it needs to execute correctly, which then allows the resources to be used by the next process.

P4: already allocated 1 1 0 0 1

P4: allocated 3 3 4 4 3 – claimed = 4 4 4 4 4 and executed.

P3: already allocated 4 1 0 2 0

P3: allocated 0 2 3 0 0 – claimed = 4 3 3 2 0 and executed.

P2: already allocated 0 1 1 1 0

P2: allocated 2 1 2 2 0 – claimed = 2 2 3 3 0 and executed.

P1: already allocated 2 0 2 1 2

P1: allocated 7 5 3 4 3 – claimed = 9 5 5 5 5 and executed.

1. This is not a safe state and will result in a Deadlock as there are no possible executions with the current quantity of available resources. A process would need to release its resources before any process is able to run.

## Question 3 - Complete

1. PTE = 512 Frame Number = (512 / 2) = 256 Address = 0011011010101010

16-bit address means there are 216 addresses in the virtual address space.

216 / page size = 29

65536 / page size = 512

65536 / 512 = page size

Page size = 128 = 27 bits

The most significant 9 bits will denote the page being referenced.

The remaining 7 bits denote the offset of the data required.

|  |  |
| --- | --- |
| Frame number | Offset |
| 001101101 | 0101010 |

Page Entry in question is (64+32+8+4+1) = Entry 109.

The Frame number is half of the Entry: 109/2 = 54.5 (Floored = 54)

The offset remains as 0101010 (32 + 8 + 2) = 42

Physical Address = 000110110 0101010

1. Segment Size = 2048 bytes (2 KB) Segment Table = 22 + 4096 + Segment Number

Logical Address = 0011011010101010

Offset = 11 bits (2^11 ==2048)

Therefore, Segment number = 5 bits

|  |  |
| --- | --- |
| Segment Number | Offset |
| 00110 | 11010101010 |

The segment in question is 6.

The segment table is 22 + 4096 + 6 = 4124 = 0001 000000011100

The offset is (1024 + 512 + 128 + 32 + 8 + 2) = 1706

Physical Address = 0001011010101010

## Question 4 - Incomplete

32-bit Physical Address 48-bit Virtual Address

1. Page Size = 8KB

48-bit Virtual Address = 248 addresses

248 / (8 \* 210)

2.8147498e+14 / 8192 = 34359738368 = 2^35

Therefore, there are 35 Pages in the table.

1. Main Memory access = 100ns Disk memory = 2ms Fault Rate = 0.005%

0.005% of the time the requested page is in Secondary Storage.

## Question 5 - Complete

|  |  |  |
| --- | --- | --- |
| Process | Burst Time | Priority |
| P1 | 3 | 3 |
| P2 | 3 | 1 |
| P3 | 4 | 3 |
| P4 | 2 | 4 |
| P5 | 7 | 2 |

1. FIFO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 |
| 0 - 3 | 3 - 4 | 4 - 7 | 7 - 11 | 11 - 13 |

Average Wait time = (3 - 3) + (4 - 3) + (7 - 4) + (11 - 2) + (13 - 7) / 5 = 14.2

Average Turnaround time = (3-0) + (4-3) + (7-4) + (11-7) + (13-11) = 13

1. RR

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | P4 | P5 | P1 | P3 | P4 |
| 0-2 | 2-3 | 3-5 | 5-7 | 7-9 | 9-10 | 10-11 | 11-13 |

Average Wait time = (10 - 3) + (3 - 3) + (11 - 4) + (13 - 4) + (9 - 7) / 5= 23.4

Average Turnaround time =

((13 - 11) + (7 - 5)) + ((11 - 10) + (5 - 3)) + ((10 - 9) + (2 - 0)) + (9 - 7) + (3 - 2) / 5 = 12.2

1. SJF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P2 | P5 | P1 | P3 | P4 |
| 0-1 | 1-3 | 3-6 | 6-9 | 9-13 |

Average Wait time = (1 - 3) + (3 - 7) + (6 - 3) + (9 - 4) + (13 - 2) / 5 = 4.2

Average Turnaround time = (1 - 0) + (3 - 1) + (6 - 3) + (9 - 6) + (13 - 9) / 5 = 9.8

## Question 6

## Question 7

## Question 8