

INDEX

[Assignment 2]



[Connor Alexander Dawkins]

[ID: DAW18175558]

[Course: BSc Computer Science]

[Module Name: Operating Systems]

[Module Code: MCOMD3PST]

[Lecturer Name: Vijay Sahota]

[Academic Year: Year 3]

[Hand-In Date: 15/01/2021]

Contents

[Task 1 3](#_Toc61006927)

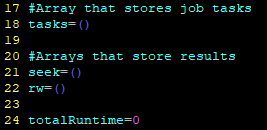
[Task 2 12](#_Toc61006928)

[References 17](#_Toc61006929)

# Task 1

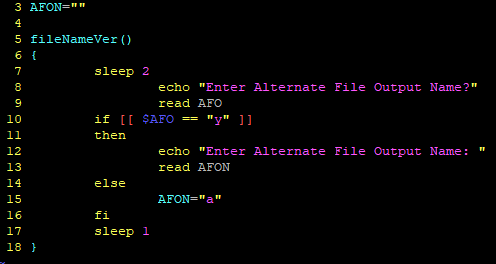
Informal Write Up

**Calculation and Writing of Simulated Results**

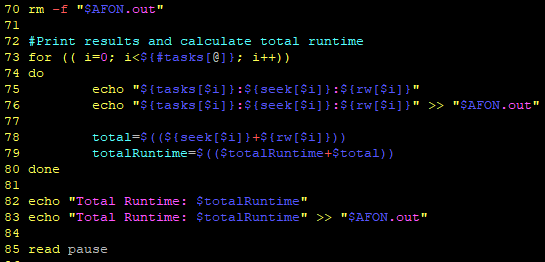


Arrays that store the results of the simulation.

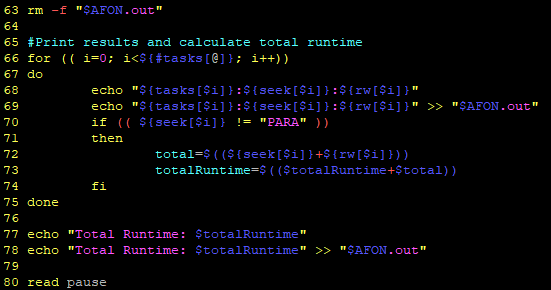




Function to take in user choice on output name

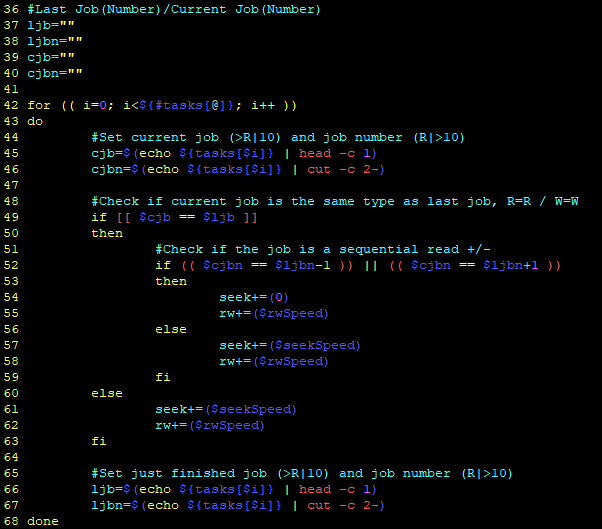


Function to display and calculate for SingleHDD



Function to display and calculate for RAID0, RAID1, RAID01, RAID100

**Correct Implementation of SingleHDD Simulation**

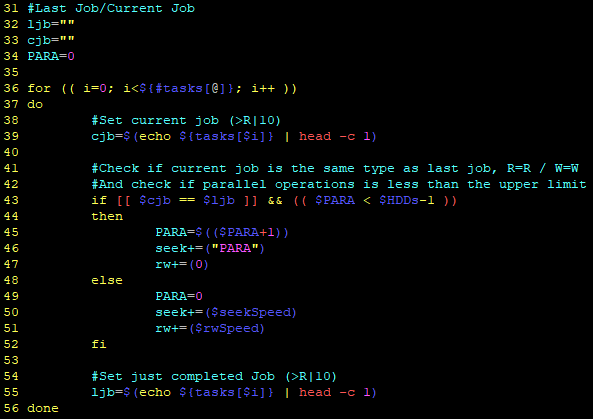


SingleHDD has a maximum Read of 1 and a maximum Write of 1, so operations cannot be run in parallel.

The code checks if the last job is both: the same as the last job (Read > Read or Write > Write) and it checks if the next operation is sequential. It does this by looking at the position of the job and checking if the last job was +/- 1 to that position. (R10 > R11 or R10 > R9)

If it was, then the reader does not need to seek to the position, so the seek time is 0.

**Correct Implementation of RAID0 Simulation**



RAID0 has a maximum Read and Write equal to the number of HDD’s in the RAID.

If both the last and current job were the same job (Read > Read or Write > Write) and the number of Parallel operations is less than the maximum allowed, which is equal to the number of HDD’s in the RAID, then the seek time is set to `PARA` and the R/W time is 0.

If the consecutive Read/Write operation exceed the Parallel limit, then it is set as a normal R/W command, requiring it to Seek and R/W normally.

Operations after this can be parallel again if conditions are correct.

**Correct Implementation of RAID1 Simulation**

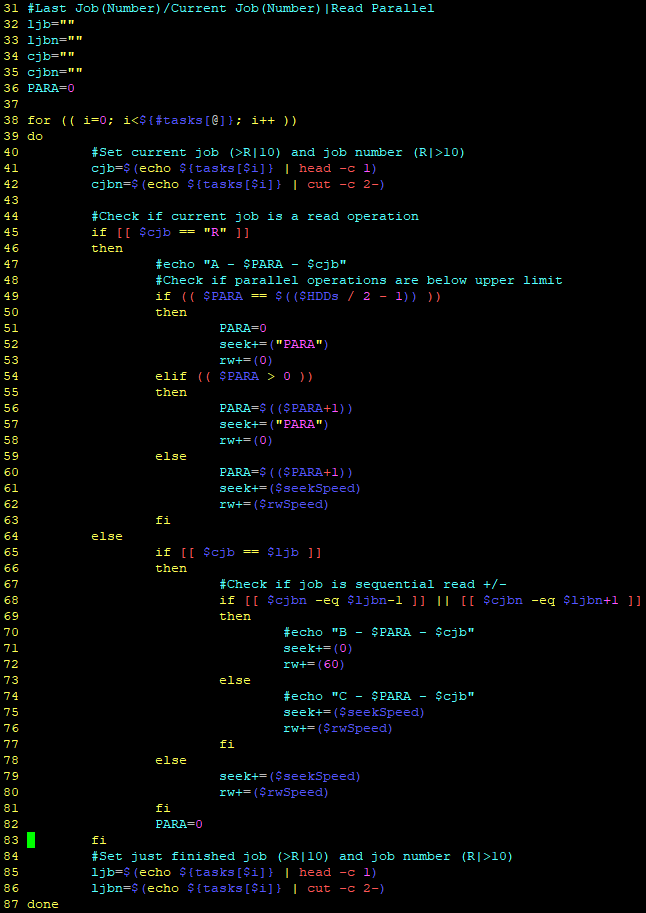


RAID1 has a maximum Read equal to the number of HDD’s in the RAID and a maximum Write of 1.

If both the current and last jobs are Read operations, then they can be run in Parallel up to the number of HDD’s in the RAID. If they exceed this number then the next consecutive job will be normal, before the Parallel stats again.

Write jobs may operate sequentially, within a range of +/- 1. (W10 > W11 or W10 > W9)

**Correct Implementation of RAID01 Simulation**



RAID01 has a maximum Read equal to half of the HDD’s in the RAID and a maximum Write of 1.

If both the current and last jobs are Read operations, then they can be run in Parallel up half the number of HDD’s in the RAID. If they exceed this number then the next consecutive job will be normal, before the Parallel stats again.

Write jobs may operate sequentially, within a range of +/- 1. (W10 > W11 or W10 > W9)

**Correct Implementation of RAID100 Simulation**



RAID100 has a maximum Read and maximum Write equal to half the number of HDD’s in the RAID.

If both the current and last jobs are Read or Write (W > W or R > R) operations, then they can be run in Parallel up to the number of HDD’s in the RAID. If they exceed this number then the next consecutive job will be normal, before the Parallel stats again.

# Task 2

Types of Operating System

**Batch**

Outline

Users of Batch operating systems do not directly interact with the computer, instead they prepare a job packet containing everything needed to perform the operation.

This is submitted to the computer operator, who will sort the jobs into batches that have similar requirements and run them in batches. (Types of Operating System - Tutorialspoint, 2020)

Named Example

FORTRAN Monitor System – IBM 709, IMB 7090 and IMB 7094 in 1957.

Use Case

As time passed the amount of time the computer spent waiting on the human operator increased, Batch Processing was used to Increase throughput of jobs on human operated computers such as tapes.

**Interactive**

Outline

Interactive operating systems allow the user to interact with the computer directly, usually through a graphical user interface and software of some kind.

Named Example

Windows, Linux or macOS.

Use Case

An interactive operating system allows flexible and powerful access to the computer that is in use. It massively lowers the barrier of entry to using a computer and enables additional software to be built on top of the operating system to increase usability and functionality.

**Hybrid**

Outline

Hybrid operating systems, otherwise known as a Hybrid Kernel, are where two, or more, operating systems are run on a single device. This can be done by dividing the memory and processor

(Hybrid Operating System, 2020)

Alternatively, multiple layers of the Kernel can be built on-top of each other, such as with Windows NT or Apple XNU, where the `User Mode` is run on top of the `Kernel Mode`, but the emulation is run on the User Mode.

(Hybrid kernel, 2020)

Named Example

Windows NT, the basis of every Windows OS from 1993 to the current day.

Apple XNU, the basis of every MacOS from 1996 to the current day.

Use Case

A Hybrid Kernel allows for the best of both worlds in terms of OS design, it approaches the speed of a Monolithic Kernel and some of the simplicity while retaining most of the safety and security of a Micro Kernel.

This is ideal for a “general” operating system, such as Windows or Mac devices.

**Real Time**

Outline

A Real Time Operating System (RTOS) is a system designed to run with very precise timing and with a high level of reliability. RTOS must have a maximum time for each critical operation, such that it can guarantee that the operation will be performed within that time limit.

(Real-time operating system, 2020)  
(What is a Real-Time Operating System (RTOS)?, 2020)

Named Example

PSOS, used in many embedded applications and systems.

VTRX, used in many avionics applications and systems.

RT Linux, an open source modification to make Linux RTOS.

Use Case

Any system that requires high accuracy of timing and reliability, such as airbag sensors and deployment, autopilot systems in aircraft; sensors, guidance and detonation of missiles.

**Embedded**

Outline

An Embedded Operating System is an OS that is designed to be run on Embedded Systems, such as a washing machine, microwave or a watch. It prioritises reliability and efficiency over all else.

They are consequently often extremely limited in their ability to perform any action outside of what they are specifically designed to be capable of.

(Embedded operating system, 2020)

Named Example

Embedded Linux, Android. Windows CE, Windows Phone. iOS, watchOS, tvOS.

Use Case

Simple devices, a washing machine, microwave or fridge might have a very rudimentary Embedded OS.

Small programmable devices, such as a watch or an mp3 player.

B

**Shortest Seek Time First**

Shortest Seek Time First (SSTF) re-orders requests as they arrive by looking at the shortest seek time between operations.

This does increase efficiency in most cases, but it can cause slowdown or even mean that some operations are never serviced if they are never the shortest seek time as operations continue to arrive.

Requests: 15, 4, 40, 11, 35, 7, 14

|  |  |
| --- | --- |
| Head Path | Tracks Travelled |
| 20-15 | 5 |
| 15-14 | 1 |
| 14-11 | 3 |
| 11-7 | 4 |
| 7-4 | 3 |
| 4-35 | 31 |
| 35-40 | 5 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 4 | 7 | 11 | 14 | 15 | 20 | 35 | 40 |

Total Time: 52

**SCAN/LOOK**

Scan/Look operates completely differently to SSTF, the head of the HDD passes in one direction and services all requests it passes over until it reaches the end of the tracks, then it turns around and goes back the other way.

This has a lower maximum efficiency than SSTF, but it provides a guarantee that every request will be served within a certain timeframe and is still significantly faster than First Come First Serve.

Requests: 15, 4, 40, 11, 35, 7, 14

|  |  |
| --- | --- |
| Head Path | Tracks Travelled |
| 20-35 | 15 |
| 35-40 | 5 |
| 40-15 | 25 |
| 15-14 | 1 |
| 14-11 | 3 |
| 11-7 | 4 |
| 7-4 | 3 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 4 | 7 | 11 | 14 | 15 | 20 | 35 | 40 |

Total Time: 56

**SCAN/LOOK over Shortest Seek Time First**

SCAN/LOOK is a superior option over Shortest Seek Time First because it offers the assurance that all requests will be looked at within a certain length of time along with maintaining most of SSTF’s speed.

The head services all operations it finds as it moves from side to side, meaning that the maximum amount of time a operation has to wait is however long it takes the head to move from the opposite side of the tracks and back.

With SSTF there is a possibility that an operation might never be serviced. For example, if the shortest seek time is in the middle, bouncing between 15-25, then there is a possibility that any operations that are close to the edge of the tracks, such as 0-5 or 35-40 are never the shortest seek time.

These might be critical operations that need to be done before a later operation, but because new operations are added to the queue and recalculated, they are never serviced by at all.

An equally important benefit is that SCAN/LOOK does not need to pre-process operations or order them at all. It is a small, but notable, burden on the processer to calculate the most efficient path between operations in SSTF, whereas SCAN/LOOK can just add the operations to the queue as they arrive.

# References

Tutorialspoint.com. 2020. *Types Of Operating System - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/operating_system/os_types.htm> [Accessed 22 December 2020].

Tutorialspoint.com. 2020. *Hybrid Operating System*. [online] Available at: <https://www.tutorialspoint.com/hybrid-operating-system> [Accessed 22 December 2020].

En.wikipedia.org. 2020. *Hybrid Kernel*. [online] Available at: <https://en.wikipedia.org/wiki/Hybrid_kernel> [Accessed 23 December 2020].

En.wikipedia.org. 2020. *Real-Time Operating System*. [online] Available at: <https://en.wikipedia.org/wiki/Real-time_operating_system> [Accessed 23 December 2020].

Ni.com. 2020. *What Is A Real-Time Operating System (RTOS)?*. [online] Available at: <https://www.ni.com/en-gb/innovations/white-papers/07/what-is-a-real-time-operating-system--rtos--.html> [Accessed 23 December 2020].

En.wikipedia.org. 2020. *Embedded Operating System*. [online] Available at: <https://en.wikipedia.org/wiki/Embedded_operating_system> [Accessed 23 December 2020].