|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **ASSIGNMENT COVER PAGE** | | | C:\Users\hoching.tay\Desktop\Lincoln_UK_06092017-01.png |
| **Programme** | | | **Course Code and Title** | | |
| UCSEW | | | COS3023/N Operating System & Concurrency | | |
| **Student’s name / student’s id** | | | **Lecturer’s name** | | |
|  | | | Siti Shafrah Shahawai | | |
| **Date issued** | **Submission Deadline** | | | **Indicative Weighting** | |
| 05.06.2023 | 4.07.2023 | | | 30% | |
| **Assignment [1] title** | Analysis of Mobile Operating System and CPU Scheduling Algorithm | | | | |
| This assessment assesses the following course learning outcomes | | | | | |
| **# as in Course Guide** | **UOWM KDU Penang University College Learning Outcome** | | | | |
| CLO1 | Describe the architecture of an operating system (OS) and its services and evaluate its implementation in a variety of scenarios. (C2, PLO1) | | | | |
| CLO2 | Discuss the process model, scheduling, IPC and synchronisation services and its behaviour in multitasking system. (C2, PLO3) | | | | |
|  |  | | | | |
|  |  | | | | |
| **# as in Course Guide** | **University of Lincoln Learning Outcome** | | | | |
|  |  | | | | |
|  |  | | | | |
| **Student’s declaration** | | | | | |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged. | | | | | |
| Student’s signature: | | | | | |
| Submission Date: 4/7/2023 | | | | | |

**TurnItIn Similarity Report**

report

**Table of Contents**

toc

**Group task distribution**

|  |  |
| --- | --- |
| **Name** | **Group task** |
| Lim Zhe Yuan | iOS history, architecture, and features |
| Saw Keat Loon |  |
| Mrigesh |  |
| Tan He Xiu |  |

**Main Report**

* **iOS**

**History**

**A person standing in front of a screen

Description automatically generated with medium confidence A picture containing text, mobile phone, multimedia, screenshot

Description automatically generated**

***Figure x****: Steve Jobs presenting about the first iPhone and comparing the efficiency of iPhone with other phone types (Millen, 2023; Wing-Girot; 2020)*

Apple’s operating system for their mobile devices, iOS, initially emerged from the market when Steve Jobs introduced the first iPhone in 2007. As version 1.0, iOS has powered the original iPhone that was said to be ahead of its time. As Newman (2022) mentioned in his article, the modern world would not be the same if Apple never released the original iPhone. At the time of its release, iPhone was not the first smartphone to surface into the market as other competitors were already releasing flip phones and devices with slide out keyboards that meet client expectations. However, iPhone was revolutionary and truly a homage to minimalism as the iPhone was designed with a flat screen surface and can be used with just one’s fingertips, allowing the user to use the phone naturally and intuitively instead of being operated with a clunky stylus or any other external pointers (Newman, 2022).

A screenshot of a cell phone

Description automatically generated with low confidence

***Figure x:*** *iOS 17 overview (Purdy & Cunningham, 2023)*

From then on, Apple had put in a lot of effort into maintaining the mobile operating system as they continue to release newer versions of the operating system annually. There have been 16 major releases to date with the latest being iOS 17. While not all releases are big enough to be considered thematic changes, each version always introduced slight improvements from its predecessor in terms of general security, privacy, or user experience (Costello, 2023). According to Costello (2023), current supported iOS versions only extends until iOS 11, while other iOS versions below version 10 have been deprecated since 2019.

**Architecture**

**A picture containing text, screenshot, design

Description automatically generated**

***Figure x:*** *iOS architecture (Redfox Security Team, 2022)*

A research on articles from the Redfox Security Team (2022) and user jagroopofficial from GeeksForGeeks (2023) shows that the iOS architecture is a layered architecture that categorizes technologies into 4 abstraction layers, which are Core OS, Core Services, Media, and Coca Touch layers. It also further packages technology interfaces into frameworks. A framework encapsulates header files, pictures, and all the dynamically shared libraries that are required for a specific technology.

The Core OS layer resides immediately atop the hardware of the device. Low-level features that form the basis of all iOS features are provided by the Core OS layer. In addition to fundamental operating system features, such as memory management, file system handling, and threads, it offers services for low-level networking and access to external accessories. It also enables iOS’s 64-bit support which facilitates the creation and performance of applications. For instance, the following low-level services are provided by this layer:

* Core Bluetooth Framework: Interaction with BR/EDR (“Classic”) and low energy Bluetooth devices.
* External Accessories Framework: Communication with devices’ accessories that are connected via Bluetooth or the Apple Lightning connector.
* Accelerate Framework: For large-scale mathematical computations and image calculations, optimized for high performance and low energy consumption.
* Security Services Framework: Controls access to your app and the data it maintains.
* Local Authorization Framework: Authenticates users using their biometrics or a passphrase they are already familiar with.

The Core Services layer presents important frameworks which helps the iOS operating system to cure itself from incapabilities and provide better functionality. It can be considered as a technological extension from the Core OS layer that make use of core OS features. Some frameworks that fall into this layer are:

* Cloud Kit Framework: Provides a medium for moving data between your app and iCloud.
* Core Data Framework: Manage the data model of a Model View Controller app.
* Core Foundation Framework: Provides data management and service features for iOS applications.
* Core Location Framework: Provide the location and heading information to the application.
* Core Motion Framework: Access motion-based data on the device.

The Media layer enables all graphics, video, and audio interaction of the system. The following shows some frameworks of this layer:.

* Core Graphics Framework: Supports 2D vector and image-based rendering. It is a native drawing engine for iOS.
* Core Animation: Helps in optimizing the animation experience of the apps in iOS.
* Media Player Framework: Enables playing playlist and iTunes library usage.
* Core Images: Provides advanced support for motionless images.
* GL Kit: Manages advanced 2D and 3D rendering by hardware-accelerated interfaces.

Lastly, the Cocoa Touch layer is the iOS architecture’s application layer which acts as an interface for the user to work with the operating system. It supports high-level applications and services of the operating system, such as touch and motion events. Examples of frameworks provided in this layer are:

* EvenKit Framework: Shows standard system interface using view controllers for viewing and changing events.
* GameKit Framework: Enables game-related data sharing online using a Game Center.
* MapKit Framework: Provides a scrollable map for general purpose use.
* PushKit Framework: Provides registration support.

**Features**

Each of the currently supported iOS versions have their own contributions and additions of features to the iOS ecosystem. Each version attempts to address prominent issues or demands from their previous versions. The following table lists out the major feature announcements that were presented for each supported iOS version and highlighted by Ava (2023):

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Features | Version | Features |
| 11 | * Personalized Contact Posters * Live Voicemail * Video voicemail and Reactions on FaceTime * Search Filter, Catch up arrow, Swipe to reply, Transcription of voice messages, Check-in features in Messages app * Live Stickers * StandBy feature * Share contact card using NameDrop * Journal app * Mindfulness feature on Health app * Create Profiles on Safari * Collaborated playlist on CarPlay * Upgraded Autocorrect and Dictation abilities | 12 | * Customizable Lock Screen * Live Activities * Better Focus Filters * New Message features * Redesigned Apple Maps * Apple Pay Later * Apple Order Tracking * iCloud Shared Photo Library * Live Texts on videos * Passkeys * Safety Check function * Improvements to Spatial Audio * Voice Isolation in phone calls * Advanced Data Protection for iCloud * Apple Music Sing karaoke |
| 13 | * New FaceTime features – Spatial audio, SharePlay, FaceTime Link, Mic modes, Portrait modes, etc. * iMessage Share with you * Better Focus Filters with Notification Summary, Signal your status, etc. * Live Text and Memory features in the Photos app * Advanced Spotlight search * App Privacy Report * Mail Privacy Protection * On-device Siri * Digital Legacy program * Redesigned Safari tabs * Detailed Maps * Improved Health and Wallet app | 14 | * Widgets on the Home Screen * App Library * Improved call notifications * Picture in Picture Mode * Redesigned Siri * Translation, Password monitoring, and Website Privacy Report in Safari * Pinned Conversations and improved group texting in Messages * New Memoji Styles and Stickers * Cycling directions and Electric vehicle routing in Maps * Translate app * App Clips |
| 15 | * System-wide Dark Mode * 30% faster Face ID unlocking * Revamped Apple Maps * Smarter HomeKit * Sign In With Apple user account * New privacy and security options * Portrait Lighting effects * New, improved Siri voice * Overhauled system apps like Photos, Mail, Reminders, and Notes | 16 | * Grouped Notifications * Screen Time * ARKit 2 for enhanced Augmented Reality * Siri improvements * Memoji |
| 17 | * iPad functions * AirPlay 2 * Support for Augmented Reality | | |

According to a Gadgets Now article (2022), iOS even has obscure features that users may not have known about. Apple offers these functionalities to users to enable advanced user customizations to adapt to most use cases in the current working environment:

* Allows users to silent all the calls received from unknown callers. With this feature, users do not need to block or ignore each unknown caller silently and can also view a list of users who called them later so that they do not miss out on any important calls.
* Move the text cursor freely, use the keyboard as trackpad. It can be challenging to place the text cursor exactly at the spot where want, especially when typing a long paragraph or sentence. To do this, press and hold on the space bar until all the letters disappear from the keyboard, then use the space as a little trackpad and place the cursor at the target location.
* Back tap on your iPhone to perform actions or shortcuts. Introduced in iOS 14, it allows users to use a shortcut to a particular app or feature when they double-tap or triple-tap on the back of their iPhones.

**Android**

* **History**

The android operating system is among the top mobile operating systems that is used by people all around the world thanks to its accessibility and benefits that it offers. Android operating system has come a long way since its creation till now.

APPSCHOPPER(2020) stated that in the year of 2003, Android operating system was created by Andy Rubin, Rich Miner, Nick Sears, and Chris White with a goal in mind to turn digital cameras into smart devices using an operating system. Such ambition did not last long as the market at that time for digital cameras was declining and the company’s attention switched to the smartphone market. APPSCHOPPER(2020) explained that in August 2005, Google acquired Android and Linux is used as the base for the Android operating system.

The usage of linux as the foundation of Android operating system enables the operating system to be used by third party mobile companies without any charges (John Callaham, 2023). Offering free usage of the Android operating system to competitors will affect Google’s yearly profit. Thus, John Callaham (2023) stated that additional services such as applications could be offered as Google felt that they could get some profit from it.

A picture containing clipart, graphics, graphic design, cartoon

Description automatically generated

Figure 1: Android logo (freepnglogos, 2019)

Figure 1 is a logo that is easily recognizable by smartphone users all around the world. The logo represents the Android operating system. dwglogo(n.d.) stated that the logo was created by Irina Blok a graphic designer from Google in 2007. Although the logo is open source, the main element remains throughout the years such as a robot having a quadtrangular torso, two antennas on a semicircular head (dwglogo, n.d.).

**Android Architecture**

In order to enable android operating systems to support various kinds of smartphones in the market, a wide range of components is needed to do so. The diagram below shows the architecture that can be found in an Android operating system.

A picture containing text, screenshot, display, number

Description automatically generated

Application layer is the top most layer in the architecture. Application layer serves as a medium where smartphone users interact with their phone most of the time thanks to its simplistic design and ease of learning. interviewbit(2022) stated that applications such as camera, clock, and galleries are pre-installed in the device. Most applications that are installed in the phone come in a square shape icon for ease of visibility.

Application framework is the second top most layer in the architecture. Application framework acts as the middle man between the application and the smartphone hardwares. interviewbit(2022) explained that during the development of Android applications, classes, interfaces and utilities are provided by the Android Application Framework. Android Hardware Abstraction Layer (HAL) is also included in the framework that mainly focuses on allowing applications to interact and control specific hardware drivers.

Platform libraries are the third top most layer in the architecture. Platform libraries layer acts as a support in developing android based applications. Native libraries such as open-source libraries like Java-based libraries and C/C++ libraries are found in the platform libraries layer. Those libraries also instruct the android powered device to handle different types of data. Examples of different data types are string, integer, or even images.

Android Runtime is the same layer as the platform library layer in the architecture. The Android runtime layer acts almost identical to the platform library layer in assisting developing android based applications. tutorialspoint(n.d.) stated that the presence of Dalvik Virtual Machine component is a kind of component that is provided from the layer that is designed for Android. tutorialspoint(n.d.) also explained that memory management and multi-threading is possible thanks to the ability of the Dalvik Virtual Machine making use of Linux Kernel via Java language.

Linux kernel is the root component in an Android system. The Linux kernel acts at the building block and the layer that enables all the functionality of an Android phone. Thompson (2023) mentioned that smartphone manufacturers are able to tweak the Linux kernel to their desired specification. The presences of multiple drivers in the layer enable developer to manage how and what does each hardware reacts to a certain situation. For example, Thompson (2023) stated that audio drivers enable the android system to receive and process the audio received from any audio component and also to create output.

**Android Features**

The competition between Apple’s IOS operating system and Android operating system has continued to grow as more and more smartphones are being sold in the market. From the limited smartphone release from Apple to the hundreds of smartphones from the android side, smartphone buyers are spoiled from choices. Some of the features that are offered from the Android side is the main reason why there exists a loyal fanbase that only uses Android powered phones.

Customizability is among the main features that is offered from Android phones. As Android is an open-source platform, smartphone makers are able to customize it to suit that own taste. This can be seen with the large number of customized Android ROMs (Read Only Memory) on the Internet. Sam(2012) stated that Custom ROM is a modified version from the pure version of Android. It is also mentioned by Sam(2012) that the source code that is obtained from Google is open source and is freely to be changed or modified by anyone. Some of the most popular custom ROMs (Read Only Memory) are CyanogenMod and Android Revolution. Some of the advantages of having custom ROMS are having full control of the device. In case something goes wrong with the current custom ROMs, the user can easily revert back to the stock operating system of the phone. The figure below shows the logo of CyanogenMod’s logo.



Figure 2: CyanogenMod ROMs (Cyanogenmodroms, no date)

Customizability also enables Android smartphone owners to change their home screen. javatpoint(n.d.) stated that there are third party launchers such as Apex, Nova that enable downloaders to add features to their smartphone such as new gestures, shortcuts or increase performance of older devices. Gestures enables smartphone users to do a specific task such as taking a screenshot or launching an application using gestures. Smartphone brand Huawei is famous for offering various gestures to their users ranging from launching the camera application by swinging their phone back and forth and taking a screenshot by knocking the screen using their knuckle.

Multitasking is also another major feature that is offered by the Android operating system. Multitasking on an Android phone is much easier than faster when compared to an Apple phone. On android, users are able to have 2 applications opened side by side of each other while in an IOS phone, they could not due to the limitation of features. tutorialspoint(n.d.) mentioned that users are able to run multiple applications and tasks at the same time. The figure below shows an example of running Facebook and Youtube applications at the same time using the split screen feature that is built into the Android operating system. Split screen can be done by dragging an application to the side of the smartphone and users are able to select another application to open beside it.

A picture containing text, screenshot, smartphone, design

Description automatically generated

**Advantages and Disadvantages of Android**

Talking about Android operating systems, it is very popular all across the world. In Google Play Store, it offers the users the chance to download a few million free applications in their own android devices and use. Android operating systems can be run on devices like tablets, smartphones, television and so on. Of course, everything in this world will have its own advantages and disadvantages. We will be discussing about the advantages and disadvantages of the Android operating systems below.

**Advantage of Android OS:**

* **An Open-Source Platform**

Android is an open-source platform because it allows user interface customization, which means the source code is available around the world. Android Apache is a licensed that is open-source operating system that every user can modify to create individual user interface (Digital Aptech, 2021). When users are developing new applications for android operating systems, they can access the main code and try to change or customize to get the outcome that users want.

* **Free Cloud Storage and Sync with own Google Account**

Android operating systems is under Google which means Android is a Google product. So users is compulsory for users to sign in their Google account when using Android devices to connect with Google cloud storage. When users successfully sign in to their account in Android devices, users will automatically have free 15GB cloud storage to store their own stuff such as photos, videos, documents and so on.

* **Freedom**

Android operating systems do not have much limit compared to IOS, because it allows user to download third-party widgets and show in their phone’s home screen. For example, if users want a unique widget on the phone, user can download through online without any restriction to make the phone look nicer (Digital Aptech, 2021).

* **Variety of Devices are Available**

There are many brands of smartphone that use Android operating systems. Examples of brands are SAMSUNG, XIAOMI, HUAWEI and so on (Digital Aptech, 2021). The users that have chosen to use Android operating systems will have a much choice of smartphones devices and different prices. Users can buy their Android smartphones depending on their budget. Every brand that runs Android on their product will have a different price range. Compare to IOS devices, there is only IPHONE can be bought and also very expensive.

**Disadvantages of Android OS:**

* **Operate Slow on Specific Devices**

The Android operating system is huge and needs a lot of space to operate. Some specific devices such as low specification devices will run slower due to the time that the user has used. The longer the time of use, the slower the android will operate. The low specification phone has slower processor and lesser storage and these are the reason why will operate slow on specific devices (Digital Aptech, 2021).

* **Security Concerns**

Due to the Android operating system’s popularity, it did not provide default virus protection for users, so Android users have a greater risk of malware and safety threats. Due to Android freedom, when users want to download third-party applications from Web browsers, they need to take cautions from that because the application might not be safe to use and might consist of virus. Not only for third-party applications, sometimes the application from Play Store also consists of viruses (Digital Aptech, 2021).

* **Background Running Application**

The devices that operate older versions of Android might be facing this issue which is the application that users open keep running in the background. Sometimes users themselves close the application but it is still running in the background for no reason(Digital Aptech, 2021). This will affect the phone’s performance and it will consume a lot of battery of the phone. For the newest version, it does not allow the application to run in background once the application has been closed, but if the application continues to run, it still consumes more battery.

* **Bad Quality of Applications**

Since Android is an open-source platform, the users or developers can just develop their own application without any cost and make money from that. Some developers will just develop an application and come with a lot of advertisements when user use the application, it is a bad experience for an user and consider as bad quality. Because of this, Android do not have quality control on every application.

**Type of Android**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version | Release Date | Feature |
| Android 1.0 | 1.0 | 23/9/2008 | -Android Market  -Web Browser  -Camera Support  -Google Features |
| Android 1.1 | 1.1 | 09/02/2009 | -Details and reviews available  -Save attachment  -Longer in call-screen timeout |
| Android Cupcake | 1.5 | 27/04/2009 | -Support third-party virtual keyboard  -Support for widget  -Video recording  -Auto-pairing  -Copy paste feature |
| Android Donut | 1.6 | 15/09/2009 | -Voice and text entry search  -Select multiple photos  -Speed improvement |
| Android Éclair | 2.0 – 2.1 | 27/10/2009 | -Support email exchange  -Bluetooth 2.1  -Improve google maps  -Support sensitive touch |
| Android Froyo | 2.2 – 2.2.3 | 20/05/2010 | -Support for the Cloud Messaging  -USB tethering  -Wi-Fi hotspot  -Support for high-PPI display |
| Android Gingerbread | 2.3 – 2.3.7 | 06/12/2010 | -Support for extra-large screen resolutions  -New audio effect  -Improve power management |
| Android Honeycomb | 3.0 – 3.2.6 | 22/02/2011 | -Support tablet holographical  -System Bar added  -Multitasking has been simplify  -Support multi-core processors |
| Android Ice Cream Sandwich | 4.0 – 4.0.4 | 18/10/2011 | -Improve visual voicemail  -Screenshot feature  -Zoom feature  -Face unlock feature |
| Android Jelly Bean | 4.1 – 4.3.1 | 09/07/2012 | -Smoother user interface  -Notification expand  -Bluetooth data transfer  -Camera improved |
| Android KitKat | 4.4 – 4.4.2 | 30/10/2013 | -Wireless printing feature  -NFC feature  -GPS feature |
| Android Lollipop | 5.0 – 5.1.1 | 04/11/2014 | -64-bit CPU supported  -Vector drawables  -Smart-lock feature  -Wi-Fi calling |
| Android Marshmallow | 6.0 – 6.0.1 | 02/10/2015 | -App Standby feature  -Support USB-C  -4K display mode  -App permissions  -Power button features |
| Android Nougat | 7.0 – 7.1.2 | 22/08/2016 | - File-based encryption  - Restricted file system  - Battery usage alert |
| Android Oreo | 8.0 – 8.1 | 21/08/2017 | -Project Treble  -Multi-display  -Google play Protect  -Autofill feature  -Automatic light and dark themes |
| Android Pie | 9 | 06/08/2018 | -New user interface  -Display cutout supported |
| Android 10 (Quince Tart) | 10 | 03/09/2019 | -Scoped storage restriction  -Biometric authentication  -Foldable phones  -Allows core OS components to be updated |
| Android 11 (Red Velvet Cake) | 11 | 08/09/2020 | -Chat bubbles  -Screen recorder  -New permission controls  -Wireless Android auto |
| Android 12 (Snow Cone) | 12 | 04/10/2021 | -Scrolling Screenshot  -One Handed Mode  -New Privacy Dashboard |
| Android 12L (Snow Cone v2) | 12.1 | 07/03/2022 | -Improvement for foldable phones, tablets, desktop and chromebooks |
| Android 13 (Tiramisu) | 13 | 15/08/2022 | -Reduce memory pressure  -Apps required permission  -Start up app improved |
| Android 14 (Upside Down Cake) | 14 | Q3 2023 | -Notification Flashes  -Security enhanced  -Satellite connectivity improved  -Health connect |

**Preferred Operating System**

Between the Android operating system and iOS operating system, the team carefully considered and preferred the Android operating system. The main reason why Android is preferred over iOS is the customizability that it offers. In the team’s opinion, it is better to have total control of one’s own phone because it shows that the phone belongs to that individual. Although iOS has come a long way from where it started, it still lacks a few key things such as notification management. Notification management on Android is far superior when compared to iOS because Android groups notifications into the same group based on the application while iOS just lays them down. Having a neat and tidy notification tray will ease the time in finding or going through notifications when there are lots of them at the time. Android phones also allow users to expand the current storage of the phone by inserting an SD card into it. Backing up images and important files into the SD card also eases the time used when migrating data to a new phone.

**Individual Task**

1. **Saw Keat Loon (0207778)**

The CPU scheduling algorithm that I have chosen is called First Come First Serve (FCFS). This algorithm executes based on requests or processes that comes first. Some of the characteristics of this scheduling algorithm are, Williams(2023) stated that it is capable of handling non-preemptive and preemptive scheduling algorithms. The meaning of preemptive scheduling algorithm is educative(no date) explained that when there is an arrival of a process with higher priority, the CPU cycle focuses on it and stops the current process. Non-preemptive is the opposite of it where there is no priority given and is purely based on who arrives first. Besides that, Williams(2023) also stated that it is easy to implement such an algorithm. The lack of rules and priority when processing requests makes this algorithm easy to be implemented in any system.

The table below shows the advantages and disadvantages of using the First Come First Serve algorithm.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| No starvation. javatpoint(no date) stated that a starvation effect happens when higher priority queue prevents lower priority queue from finishing their turn in the CPU. The First Come First Serve algorithm prevents that because it does not have any priority rules. | Convoy effect. geeksforgeeks(2019) stated that a convoy effect happens when there are a few long processes being processed in the CPU hence slows down the remaining processes. The convoy effect can occur in the First Come First Serve environment when there are many long requests in a queue. |
| Fair algorithm. First Come First Serve practices works by processing which task or request that comes first. This will cause all requests or tasks to have a fair and square treatment ignoring their duration and priority. | Not compatible with a time sharing system. A time sharing system requires processes to be done in a quick or low average waiting time. In a First Come First Serve environment, the average waiting time is high due to chances of having a long processing time request in the front and short processing time request at the back of the queue. This will cause all the short processing requests to wait for the long processing request to be released from the CPU. Hence it will increase the average waiting time. |

To properly understand how a First Come First Serve algorithm works, the diagram below shows an example of how it starts and ends.

|  |  |  |
| --- | --- | --- |
| Process ID | Arrival Time | Burst Time |
| P1 | 0 | 3 |
| P2 | 1 | 2 |
| P3 | 2 | 4 |
| P4 | 2 | 5 |
| P5 | 3 | 3 |

The table above shows the process IDs, arrival time and burst time of each process. The arrival time shows the time when each process reached the CPU. As this algorithm works based on who comes first, the gantt chart below shows the queue for the processes based on their arrival time.

A picture containing line, screenshot, rectangle, parallel

Description automatically generated

As P1 arrives at 0, thus, it is the first in the queue. Then following it is P2 where its arrival time is at 1. Following P2 is P3 and then P4. P3 is in front of P4 because it arrives first in the table although they have the same arrival time. P5 is the last in the queue as it arrives the latest.

A picture containing text, screenshot, font, white

Description automatically generated

The formulas in the diagram above will be used to find the turnaround time for each of the processes and their waiting time. To find the average waiting time, it is needed to find all of their waiting time and divide by the total number of processes. Rouse (2017) stated that turnaround time is time needed from the arrival of a process till the completion of it. Admin AfterAcademy (2019) explained that waiting time is the time needed or spent by a process waiting in the ready state for the CPU.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Process ID | Arrival Time | Burst Time | Turnaround Time | Waiting Time |
| P1 | 0 | 3 | 3 - 0 = 3 | 3 - 3 = 0 |
| P2 | 1 | 2 | 5 - 1 = 4 | 4 - 2 = 2 |
| P3 | 2 | 4 | 9 - 2 = 7 | 7 - 4 = 3 |
| P4 | 2 | 5 | 14 - 2 = 12 | 12 - 5 = 7 |
| P5 | 3 | 3 | 17 - 3 = 14 | 14 - 3 = 11 |

The table above shows the calculated turnaround time and waiting time for each of the processes. For example, to obtain the turnaround time for process P1, the completion time, 3 subtracts the arrival time, 0 and 3 is obtained. While to obtain the waiting time, the turnaround time,3 that is calculated is used to subtract the burst time which is 3. Thus, its waiting time is 0. The formulas from the diagram above are used throughout the processes to obtain each of their turnaround and waiting time.

A picture containing text, screenshot, display, software

Description automatically generated

The figure above shows the method to obtain the average waiting time for the given processes by using the calculated waiting time for each process and dividing them with the total number of processes. From the final answer, the calculated value is 8.

A picture containing text, screenshot, display, software

Description automatically generated

The figure above shows the method to calculate the average turnaround time for the processes. To obtain the average, each of the calculated turnaround time is summed up and is divided by the total number of processes. Thus, the calculated average turnaround time is 4.6.

1. **0204677 Lim Zhe Yuan**

**Shortest Job First (SJF)** is a type of scheduling algorithm in the operating systems where the processor executes the processes that have the shortest execution time, or burst time, first. It is one of the approaches of long-term scheduling (Aman, 2022) where the main goal is to regulate programs, select processes from queue efficiently and load them into memory for execution (Williams, 2023). Burst time is an important factor in SJF scheduling, and it can be defined as the time required by a process to finish executing (Aman, 2022). In case of a tie in burst time, it is broken by the FCFS scheduling algorithm (Singhal, 2018). According to Ishaque (2023), the algorithm can be implemented in 2 ways: preemptively and non-preemptively.

* **Preemptive SJF**

Preemptive SJF is also called Shortest Remaining Time First (SRTF). It works in a way where it pays close attention to the arrival times of incoming processes. If another process arrives with an even shorter burst time ﻿after the process with the shortest burst time and arrival time is given CPU time﻿, the CPU cycle is allocated to the process with the shorter burst time (Ishaque, 2023). Consider the following example table of processes with their arrival times and burst times:

|  |  |  |
| --- | --- | --- |
| Processes | Arrival time | Burst time |
| P1 | 1 | 3 |
| P2 | 0 | 9 |
| P3 | 1 | 2 |
| P4 | 5 | 4 |
| P5 | 3 | 6 |

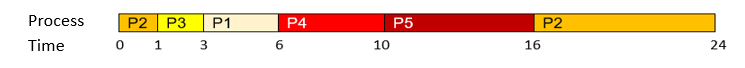
As shown in the table above, P2 would ideally be allocated to the CPU and executed first as it is the first process to arrive to the processor. However, after the first second of execution, P1 and P3 would arrive next with even shorter burst times than P2. Therefore, preemptive SJF algorithm takes place by suspending the job execution of P2 and instead allocates P3, the process which has the shorter burst time among P1 and P3, to be executed by the processor instead. In other words, context switching takes place until all processes have arrived to the waiting queue, and then the CPU executes them based on their remaining burst times that may have been reduced because of earlier executions of processes that had undergone suspension.

Specifically, the way of arranging and calculating how the processes will be scheduled is determined by considering how the arrival times of new processes affect the remaining burst time of ongoing processes and the arrived processes in the waiting queue. For example, the burst time of P2 will be reduced by 1 after the first second and becomes 8, while both P1 and P3 arrives into the waiting queue with shorter burst time at the same time. Therefore, it can be determined that the CPU would prioritize P3, which has the shorter burst time than P1, instead of P2 at the next second. This logic is then applied for every second of the processor’s execution to generate the full processor timeline. The following figure shows the method to recalculate the remaining burst time of P2 which was affected because of the arrival of P1 and P3 at the first second, and is also the method used to recalculate the rest of the processes down the timeline.

A screenshot of a computer

Description automatically generated with low confidence

In the case of an idle schedule state, the CPU selects the available processes that have arrived at that point of time with the shortest burst time. The following Gantt chart shows the complete timeline of the above process scheduling after all processes have been completed.

 Based on the Gantt chart above, it can be theorized that it would take 24 time units to complete all 5 processes, and P2 is divided into 2 execution segments to give way to processes that have shorter burst times. It can also be concluded that processes that arrive early, such as P1 and P5, will not be immediately selected for execution as well.

* **Non-preemptive SJF**

Non-preemptive SJF works slightly different from preemptive SJF. Although it still pays close attention to process arrival times and prioritizes shorter burst times, non-preemptive SJF does not have the ability to suspend ongoing processes in the CPU. Once a process has been removed from the waiting queue and given CPU time, it will execute until it is either completed or terminated (Ishaque, 2023). This behaviour can be illustrated by using the previous example for preemptive SJF:

|  |  |  |
| --- | --- | --- |
| Processes | Arrival time | Burst time |
| P1 | 1 | 3 |
| P2 | 0 | 9 |
| P3 | 1 | 2 |
| P4 | 5 | 4 |
| P5 | 3 | 6 |

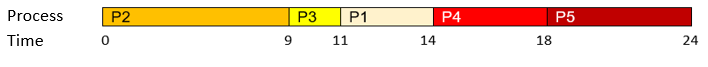
As shown in the table above, P2 would be allocated again to the CPU and executed first as it is the first process to arrive at the processor. However, contrary to preemptive SJF, instead of relinquishing control of CPU to incoming processes that have shorter burst times than the executing process itself, the CPU continues to execute the entirety of the process equal to the amount of its burst time until its completion. No suspension occurs when there is an arrival of new processes that have shorter burst times and these processes must wait for the current executing process to complete before it can be selected as the next execution target. When a process completes, the CPU evaluates the available processes that have arrived at that point of time and selects the process with the shortest burst time.

Like preemptive SJF, the way of arranging and calculating how the processes will be scheduled is also determined by considering the arrival times of new processes, burst time of ongoing processes and the waiting queue. However, the burst time for processes never changes due to job segmentation, which makes time calculations and process selection easier. This logic is then applied again every time the processor finishes executing a process to generate the full processor timeline. The following figure shows the happenings of the waiting queue and the CPU within the first 9 seconds of the schedule which is used to construct a part of its Gantt chart.

A screenshot of a computer

Description automatically generated with low confidence

The following Gantt chart can be constructed using this logic for other processes as time moves forward, which shows the complete timeline of the above process scheduling after all processes have been completed.



As shown in the Gantt chart, as P2 arrives well before other processes, it is selected and executed first by the CPU. Even after the arrival of other processes that possess shorter burst times during P2’s execution, the CPU continues to execute P2 until it is completed. Then, P3 is only executed next after P2 is completed as it has the shortest burst time among the other 4 processes. In this example, the whole process also took 24 time units to complete like in preemptive SJF.

**Advantages**

According to a Studytonight article (2023), the SJF scheduling algorithm is the best approach to minimize waiting time for processes compared to a first come, first served approach. In hindsight, this characteristic should only apply to preemptive SJF as it effectively prioritizes shorter processes, whereas non-preemptive SJF may put these processes on hold for a prolonged period of time because of an active execution in the CPU. This was illustrated in the previous example for non-preemptive SJF, where P5 is held until the end before executing even though it arrived very early at the start. However, it can be said for both algorithms that it is optimal in terms of minimizing the average waiting and turnaround time for processes (Ishaque, 2023; Singhal, 2018). SJF scheduling algorithm can also be optimal if all the processes are available at the same time. This means that if the arrival time of all processes is the same, the algorithm is ensured to produce an optimal result that maximizes the reduction of waiting time (Studytonight, 2023).

SJF scheduling algorithm is also useful for batch system processes where burst time is known beforehand. Batch systems are systems that perform huge repetitive jobs that are done without direct user interaction. They are able to receive jobs from multiple users and can keep the number of received jobs in memory. It was mentioned that these systems process jobs in a first-come, first-served manner (JavaTpoint, 2023). Therefore, with the introduction of SJF scheduling algorithm, batch systems can gain the ability to prioritize shorter tasks to save waiting time and result in a better user experience.

**Disadvantages**

There are chances that this algorithm will face starvation of processes that arrive later with larger burst times (Ishaque, 2023). Starvation is a resource management issue where the process does not have resources (in this case execution time) because it is being used by other processes (Mansi, 2022). if there are too many processes in the queue that have shorter burst times, then it will cause longer processes to wait for a prolonged time in the queue (Aman, 2022).

It was also mentioned that process burst times should be known beforehand to be able to use SJF scheduling algorithm. As it was previously established, the algorithm emphasizes on the use of process burst times. Without burst times for any process, it is impossible for the CPU to decide when selecting the processes as it does not know which process has the shorter burst time. This also makes SJF scheduling algorithm useless for implementing short term scheduling, or CPU scheduler, because of the inability to predict burst times of the contemporary processes (Ishaque, 2023; Williams, 2023).

1. **Tan He Xiu (0207737)**

Scheduling algorithm is an operating system which connected with the hardware of computer to user by enable the communication between the software application and the devices that has been connected (unstop.com., n.d.). Scheduling algorithm is responsible for carrying out the tasks for humans such as managing memory, files, processes and so on. This can highlight as how the operating system run many tasks and cover up several processes or requests at once. There are many types of scheduling algorithms that I have learned in class which are First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR) and Priority. In this report, I will be going to share about the Shortest Job First (SJF).

SJF or we can also know as Shortest Job First. It is an algorithm which selects the process that have the least execution time for the afterwards execution. For this algorithm method, which is shortest job first, it can be preemptive or non-preemptive. Shortest Job First basically is to minimize on the average amount of time for the next processes that waiting to be executed (Williams, L., 2023). It is also used as a measure of completion time for each task or process. Next, in the process that shorter tasks continue, starvation might be the outcome. To prevent starvation, the concept of ageing can be used to overcome this issue. To have a shorter turnaround time, the task should be completed first and it will increase the job output (Williams, L., 2023).

There are some advantages for the Shortest Job First algorithm. The first advantage is SJF is used for the long-term scheduling compared to other scheduling algorithms. Shortest Job First has decreased the average time of waiting over the First In First Out algorithm. For the jobs that are executed in batch is suitable for Shortest Job First and for those whose run time is established. A task’s burst time estimation can be determined from the work description through the batch system of long-term scheduling (GeeksforGeeks., 2020).

Talking about advantages, Shortest Job First also has its own disadvantages. The first disadvantage is it is hard to predict the job completion time because it must be known earlier. Next, Shortest Job First cannot be implemented that relate to short term CPU scheduling. The reason why cannot be implemented is because there is no correct way to predict the upcoming CPU burst length. Other than that, it may cause starvation and not decrease any of average turnaround time (GeeksforGeeks., 2020).

**Example of Preemptive demonstration:**

**A picture containing text, blackboard, handwriting, menu

Description automatically generated**

Picture 1: Preemptive

To calculate the average waiting time, we must do some calculations using the method of preemptive SJF. Before I explain the steps of preemptive SJF, I will show how to calculate the average waiting time. To get the average waiting time, we must calculate the waiting time of each process queue which is Turnaround time minus burst time. Burst time is given by the question but turnaround time do not, so we must find turnaround time by ourselves. The way that finds the turnaround time is completion time minus arrival time. Arrival time is given by question and completion time will be found in the gantt chart which later I will explain on it. Next, arrival time is the time that the process that can start to execute. Burst time is the time that the process needs to be taken to finish. Turnaround time is the amount of time that have to take to complete a process and completion time is the time that the process has finished executing.

From the above example of my phone screenshot, P4 has 0 arrival time which becomes the first to be executed. P4 can run finish the process which is 3ms even though P3 arrival time is 1 and P1 arrival time is 2 because P4 has less burst time compared to P3 and P1. Next, we need to compare P3 and P1 who has the least burst time and execute it. P1 has the least burst time so P1 can execute for 1ms. It only runs for 1ms because P5 has arrived at 4ms. Since P1 has run for 1ms so left 5 burst time and compare to P5 burst time who has the least burst time. In result, P5 has lesser burst time so we run P5 for 1ms. Same thing, P2 arrives at 5ms so need to compare again and P2 has lesser burst time and we run P2 first. Since P2 is the last process, we can run finish the burst time which is 2ms. After that, we run back P5 process that is left just now which is 3, followed by P1 which is 5 and lastly is P3 which is 8. At last, the total time will be 23ms.

Now the gantt chart is completed and we have all the information we need to calculate Waiting Time. The formula is turnaround time(Completion time – arrival time) minus burst time. Calculate all the processes and add up together and divide by 5 then we can get the final answer of Average Waiting Time which is 4.6ms.

**Example of Non-Preemptive demonstration:**

**A picture containing text, blackboard, handwriting, font

Description automatically generated**

Picture 2: Non-Preemptive

For non-preemptive SJF, it is slightly different compared to Preemptive SJF. It is simpler and not complicated than preemptive SJF. I will the same example of preemptive SJF to explain the steps. The first to execute is P4 because of the 0 arrival time. It executes 3ms which follow the burst time given. Next, it is 3ms, which mean P3 and P2 have arrive, both need to compare who have the least burst time and choose to execute. P1 has the least burst time so P1 execute after P4. When it executes, it runs for 6ms and the total of time is 9ms. When it comes to 9ms, all processes have arrive and what we need to do is just compare which processes have the least burst time and execute it. P2 have the least burst time so P2 will be execute, followed by P5 and last will be P3. Total time will be 23ms. For the calculation of Average waiting time is same as preemptive.

**References (Groupwork)**

APPSCHOPPER (2020) *Android OS.* Available at: <https://www.appschopper.com/blog/history-evolution-googles-android-os/> (Accessed : 25 June 2023).

Callaham, J (2023) *The history of Android: The evolution of the biggest mobile OS in the world*. Available at: <https://www.androidauthority.com/history-android-os-name-789433/> (Accessed: 25 June 2023).

Digital Aptech. (2021). Advantages and Disadvantages of Google Android. [online] Available at: <https://www.digitalaptech.com/advantages-and-disadvantages-of-google-android/> (Accessed: 26 June 2023).

Hill, S. (n.d.). The 12 Top New Android 14 Features. [online] Wired. Available at: <https://www.wired.com/story/how-to-download-android-14-and-features/>. (Accessed: 26 June 2023).

InterviewBit (2022) *Android Architecture – Detailed Explanation*. Available at: <https://www.interviewbit.com/blog/android-architecture/> (Accessed: 25 June 2023).

Sam (2012) *So, what is a custom ROM?*. Available at: <https://www.androidauthority.com/what-is-a-custom-rom-android-74072/> (Accessed: 25 June 2023).

Thompson, B (2023) *Android Architecture: Application Layers, Framework, Component*. Available at: <https://www.guru99.com/android-architecture.html> (Accessed: 20 June 2023).

Tutorialspoint (no date) *Android - Architecture*. Available at: [https://www.tutorialspoint.com/android/android\_architecture.htm#](https://www.tutorialspoint.com/android/android_architecture.htm) (Accessed: 20 June 2023).

Tutorialspoint (no date) *Android - Overview*. Available at: <https://www.tutorialspoint.com/android/android_overview.htm> (Accessed: 21 June 2023).

Wikipedia. (2020). Android version history. [online] Available at: <https://en.wikipedia.org/wiki/Android_version_history>. (Accessed: 26 June 2023).

Individual

Admin AfterAcademy (2019) *What is Burst time, Arrival time, Exit time, Response time, Waiting time, Turnaround time, and Throughput?*. Available at: <https://afteracademy.com/blog/what-is-burst-arrival-exit-response-waiting-turnaround-time-and-throughput/> (Accessed: 1 July 2023).

Educative (no date) *What is preemptive priority scheduling algorithm?*. Available at: <https://www.educative.io/answers/what-is-preemptive-priority-scheduling-algorithm> (Accessed: 26 June 2023).

Geeksforgeeks (2019) *Convoy Effect in Operating Systems*. Available at: <https://www.geeksforgeeks.org/convoy-effect-operating-systems/> (Accessed: 27 June 2023).

Javatpoint (no date) *Starvation and Aging in Operating Systems*. Available at: <https://www.javatpoint.com/starvation-and-aging-in-operating-systems> (Accessed: 27 June 2023).

Rouse , M (2017) *Turnaround Time.* Available at: <https://www.techopedia.com/definition/23798/turnaround-time-tat> (Accessed: 1 July 2023).

Williams, L (2023) *FCFS Scheduling Algorithm: What is, Example Program*. Available at: <https://www.guru99.com/fcfs-scheduling.html> (Accessed: 26 June 2023).

**References (Individual work – Keat Loon)**

**References (Individual work – Zhe Yuan)**

**References (Individual work – He Xiu)**

**References (Individual work –Mrigesh)**

----- Please refer to the attached excel file for the marking rubric -----