**A2 | Operating System:**

* **Operating System:** - the low-level software that supports a computer’s basic functions.
* **Real-Time Operating System:** - is an operating system intended to serve real-time application that process data as it comes in, typically without buffer delays. For example, ‘Air Traffic Control Systems’.
* **Single-User (Single Task) Operating System: -** used for systems that will be used by only one user and complete only one task at a time used for simple/less powerful devices. It only has to deal with one person at a time and running one application at a time. For example, ‘The Palm OS’ & ‘Windows 95’
* **Single-User (Multi-Tasking) Operating System: -** is an operating system that allows a single user to simultaneously run multiple applications on a computer. This is the type of operating system that is found in personal desktop and laptop computers. For example, ‘Microsoft Windows’.
* **Multi-User Operating System: -** is an operating system which allows multiple users that are on different computers to access a single system’s OS resources simultaneously. It allows more than one user to access programs and data at the same time. Typically used on servers, mainframes and supercomputers. For example, ‘UNIX Operating System’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Real-Time Operating System:** | **Single-User (Single Task) Operating System:** | **Single-User (Multi-Tasking) Operating System:** | **Multi-User Operating System:** |
| **Advantages:** | - Processes Data in Real Time  - Maximum use of devices and systems thus gives more output from all the resources  - Time given for shifting tasks is very less  - Error free  - Memory allocation is well managed | **-** Supports one user at a time. This means that no other user will interfere with running applications  - Easy to maintain as the system uses less resources and complexity of them is less so it is easy to maintain and debug.  - Less chance of damage. These systems include less requests to hardware/software so they have a less chance to become damaged compared to multi-user operating systems  - Concentrates on one tasks. | **-** Supports one user at a time. This means that no other user will interfere with running applications  - Time sharing. All tasks are given a suitable amount of time and no waiting time occurs for the CPU  - Programs can run in the background  -User can use multiple programs at the same time | **-** Printing jobs in an office is best handled by multi operating system.  -Each user can access the same document on own PC. For example, if one computer has song then the other computers attached with it will play that song. Airlines use this system to handle ticket reservations.  -If one computer in the network encounters an error then the other computers will not be affected. |
| **Disadvantages:** | **-** Only some task run at the same time  - Sometimes the system resources are not good enough and they are costly as well  - Complex and difficult to write algorithms are used  -It requires specific device drivers  - They are very less prone to switching between tasks | **-** Idle time is higher. If only one task is running and this task don’t require a lot of memory, then these devices remain idle.  - tasks take longer time to complete. As no multiple tasks can be ran at a time, then many tasks are waiting for the CPU to be done with the previous task | **-** Limitation of memory. When a computer user opens many programs at a time, the computer becomes slow due to many programs being loaded into main memory and the CPU cannot give good time for each program. The response time of completing a job becomes higher. Computers with low RAM suffer from this problem.  - Limitation of the processor. If the processing speed of a CPU is slow, then processing and managing multiple programs will be slow.  -CPU heats up. CPU is constantly busy. Cooling equipment is needed to prevent over heating | **-** If a computer contains private information then sharing your computer with this operating system may be dangerous  -If one computers become infected with malware then other computers can also suffer.  -Your computer information will be shared to other users |

**A2 | The role of the Kernel in controlling and managing system components and tasks:**

* **Kernel: -** This is a central core component of an operating system. It manages the operations of the computer and the hardware, most notably memory and CPU time.
* **Micro-Kernel: |** runs services those are minimal for operating system performance
* **Monolithic-Kernel: |** an operating system architecture where the entire operating system is working in kernel space.
* **Hybrid (Macro) Kernel:**  a combination of both micro-kernel and monolithic-kernel to have the best of both worlds.

|  |  |
| --- | --- |
| **Micro-Kernel:** | **Monolithic Kernel:** |
| * It can be broken down into separate processes called servers. | * Provides CPU scheduling, memory management, file management and other operating system functions through system calls. |
| * In order to write a microkernel, more code is required. | * It is a single large process running entirely in a single address space. ( the entire operating system runs as a single program in kernel mode) |
| * If a service crashes, it never affects the working of a microkernel. Server malfunction is also isolated as any other user program's malfunction. | * It is a single static binary file. For example: ‘Unix’ & ‘Linux’. |
| * Servers communicate through IPC. | * If anyone of the services fails, then it leads to entire system failure |
| * The architecture is small and isolated therefore it can function better. | * If a user has to add any new service, then the user needs to modify the entire operating system. |
| * Microkernels are modular, and the different modules can be replaced, reloaded, modified without even touching the Kernel. | * Security issues are always there because there is no isolation among various servers’ present in the kernel. |
| * Increased security and stability will result in a decreased amount of code which runs on kernel mode. | * In order to write a monolithic kernel, less code is required. |
| * The performance of a microkernel system can be indifferent and may lead to some problems. |  |

1. **Program Execution:** | To load software, allocate resources, run and terminate when finished.
2. **Interrupts: |** This is an event external to the currently executing program on the CPU. For example, components such as the mouse to request servicing by the CPU. When these events happen, the CPU will be executing some random program, which may be completely unrelated to the event. The hardware device associated with the event (for example a keyboard) informs the processor so that it can be processed. This is called an Interrupt. This interrupts the processor from its current execution to inform the processor about a pending event.
3. **Modes: |** To switch between user modes, to run apps, and system (kernel mode), to run low-level code that the operating system needs for operations such as reading memory.
4. **Memory Management: |** Allocating RAN to running programs and to reuse RAM when programs end.
5. **Multi-tasking: |** To allocate enough CPU and peripheral time to every running program so that they all work concurrently.
6. **Disc Access: |** To read from or write to backing storage
7. **File Systems: |** This controls how data is stored and organized on backing storage. Operating systems use multiple of different systems, which use different addressing methods, in order to allow the operating system to save and retrieve files. For example, a simple file system called ‘FAT32’ are used for low capacity devices such as USB flash drives, where security is not required. However, more sophisticated files systems such as ‘NTFS’ can address larger devices and provide additional features such as user access rights, improved ability to recover from disc errors and encryption.
8. **Device Drivers: |** This allows the operating system to communicate with hardware components. For example, a ‘Graphics Card’. This will allow it to use its best features and improve performance.

**A2 | The role of the operating system in managing networking and security:**

* **At Home: |** Networking within a small environment, such as a home setup, will be most likely be peer-to-peer. This involves the use of no servers along with the basic sharing of the internet and printing. The security on a peer-to-peer network is likely to consist of a power-on password or user recognition for each computer and a WPA2 password for joining the Wi-Fi and passwords on shared folders.
* **In an Organisation: |** Networking within a large environment, such as an organisation, will most likely be a client-server setup., with servers controlling the resources that user accounts can access. Security on a client-server network is based upon authentication of users when they log in, this is done by typing their ID and password, and then upon the parts of the network they have right to access. Systems running on the network often have separate authentication log-ins.

**A2 | Factors affecting the choice of operating system:**

|  |  |
| --- | --- |
| **Stability and Robustness:** | The operating system must not be prone to crashes or damages. For example, some operating systems require to be rebooted occasionally. It must be reliable. |
| **Compatibility:** | The operating system must be compatible with whatever you are trying to do. |
| **Memory Management:** | The operating system must be efficient at managing memory otherwise the performance of the operating system can be poor. |
| **Memory Leaks:** | Some operating systems suffer from memory leaks, which is when some process requests a chunk of memory for temporary storage, but then does not subsequently release it |
| **Cost and Support:** | An organisation may have a limited budget on how much they can spend. Also, they may want an operating system with available support if issues arise. |

**A2 |Factors affecting the choice and use of user interfaces:**

* **Graphical User Interface (GUI): |** A form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators.
* **Command Line Interface (CLI): |** This processes commands to a computer program in the form of lines of test.
* **Text-Based/Menu-Based Interface: (TUI): |** A user interface that uses text, symbols and colours.

|  |  |  |
| --- | --- | --- |
| **Graphical User Interface:** | **Command Line Interface:** | **Text-Based Interface:** |
| * Intuitive (Easy to use) – This means it can be used by a range of different people | * The user has to know the commands | * Intuitive ((Easy to use) – This means it can be used by a range of different people) |
| * Does not require prior knowledge to use | * Commands usually have to be entered in full | * Does not require much prior knowledge to use. Training may be needed depending on the context |
| * Often uses images, which require more memory & processing time/power | * Interface can be daunting (difficult to use) | * It does not use images, only text. This means it uses a lot less memory and less processing time/power |
| * Visually appealing & User friendly | * User most likely to make mistakes | * Quicker compared to a GUI |
| * Users are more restricted on the use of menus | * No graphics/no menu | * Not necessarily visually appealing |
| * Pointing devices can be used to select items | * The user has complete control |  |
| * Can be slow | * No pointing device is used |  |
| * Spelling and typing errors can be avoided | * quicker |  |

**A2 | Utility Software:**

* **Utility Software**: - Software that is designed to perform a specific task to extend or aid the operating system.

|  |  |
| --- | --- |
| **Anti-Virus/Anti-Malware:** | * Used to prevent, detect and remove malware/viruses |
| **Backup Software:** | * Software that is used to perform a backup. This is the creation of supplementary exact copies of files or databases. |
| **Compression Tools:** | * Compression is the method used to make files smaller by reducing the number of bits. This is software that can compress and decompress various file types. |
| **Disk Analysers:** | * a diagnostic tool that checks the condition of a computer’s hard drive and reports on available space. |
| **Disk Defragmenters:** | * reorganises data on a hard disk drive (HDD) so related data is grouped together in order to improve search and load times |
| **Disk Partitioners:** | * used to split a single storage medium into multiple volumes. Allows a user to separate data and use multiple file systems on a single disk. |
| **Encryption Software:** | * – a method of protecting data by scrambling the contents using algorithm (which makes use of a key) so that data cannot be read unless the correct key is provided. Encryption can be used during data transmission or to protect stored data. |
| **File Managers:** | * A program used to organize, list and locate files and directories on a computer |
| **Firewall:** | * Protects a network or system from unauthorized access with a firewall |
| **Network Utilities:** | * These are software utilities that are designed to analyze and configure various aspects of computer networks. For example, ‘Ping’. The PING utility tests connectivity between two hosts. |
| **Package Managers:** | * A collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer’s operating system in a consistent manner. |

**A2 | Application Software:**

* **Application Software: |** a program or a group of programs designed for end users, which are people who actually uses a particular product.

|  |  |
| --- | --- |
| **Communication Software:** | * includes email, instant messaging, social media and video conferencing. For example: ‘Snapchat’ & ‘Outlook’ |
| **Computer Aided Design: (CAD)** | * software used to produce highly detailed technical drawings such as building plans or when designing a car’s engine. For example: ‘AutoCAD’ |
| **Database Management Systems:** | * used to create and manage complex, relational databases. For example ‘InterBase’ |
| **Digital Graphics and Animation:** | * the art of creating of moving images by the use of the computers. For example ‘Adobe Photoshop’ & ‘Blender’ |
| **Enterprise Resource Packages: (ERP)** | * an integrated software package used in industry to share data between all areas of a business including managing orders, stock, payroll and company finances |
| **Entertainment Software:** | * a group of software programs that include, media players and games. |
| **Office Software:** | * Powerpoint, Word, Spreadsheet etc |

**A2 | Factors affecting the choice, use and performance of Hardware, Utility Software & Application Software:**

|  |  |
| --- | --- |
| **Security:** | * The state of being free from danger or threat |
| **Performance:** | * The action or process of performing a task or function |
| **Efficiency and Effectiveness** | * The degree to which something is successful in producing a desired result |
| **Compatibility:** | * A state in which two things are able to exist or occur together without problems or conflict |
| **Productivity:** | * The measures of the efficiency of production |
| **Requirements & User Needs:** | * Does it fit within the guidelines of the user’s desires? |
| **Implementation Timescales & Testing Migration:** | * How long it will take to implement based on the time frame given to have it operating and ready to use. Plus, how long will it take to test? |
| **Cost & Budget:** | * How expensive will it be and will it fit within the individual’s budget? |
| **Ease of Use & User Experience:** | * It is easy to use? Intuitive? Does it make a positive user experience? |

**A2 | Open Source Software:**

* **Open Source Software:** - This is software released where the source code of a program can be accessed and modified to the user’s will.

|  |  |
| --- | --- |
| **Advantages of Open Source Software:** | **Disadvantages of Open Source Software:** |
| * Usually Free. Cost of buying is zero or very low – although payment may still be needed for installation or training | * Small projects may not get regular updates |
| * Software will not be dropped for commercial reasons | * Can be buggy |
| * Often made for the greater good, not profit. This is because it benefits everyone and encourages collaboration. | * May have limited or no user documentation (user documentation is designed to assist users to help them use the product) |
| * Flexible. Software can be adapted by the user to fit their needs | * No warranties if something goes wrong |
| * Open Source has a wider pool of collaborators compared to Proprietary Software. As a result, the community programmers can be more creative and innovative than the programmers of one company | * No customer support. It does not come with extensive (However the community can make up for this) |
| * Popular software is often very reliable of being secure as any problems are quickly solved by the community. Transparent bugs are openly acknowledged and dealt with. | * Might not be as user-friendly as commercial versions |

|  |  |  |
| --- | --- | --- |
| **Implications of Open Source Software:** | **Software Cost Benefits:** | **Software Development Benefits:** |
| May not work with existing systems, or may require a complete changeover in order to make it compatible | Lower costs of marketing and distribution | More rapid development |
| Installation, customization, support and training costs may be high | Lower costs of developer teams | Loyal teams – sense of ownership |
| Help from user community may not as fast nor as tailored | Lower cost of management | Build team skills – all code assessed transparently by peers |
| Can evolve to suit developer’s needs, rather users’ needs | Commercial companies are able to build on solid, reliable code | Flexible, quality, modular development – individuals with better skills can work on what they are best at |
|  |  | Innovative – focus on technical rather than commercial objectives. |