Computer System and Data Structures

The computer system is a set of components that perform a specific function or task. In computer science, computer systems refer to hardware components such as the CPU, memory, and storage devices, or they may refer to software components such as an operating system, a database management system, or an application (Uri.edu, 2019).

A data structure is the organisation of data to reduce storage space and difficulty while performing various tasks. Data structures handle and work with large amounts of data in multiple fields, such as database management and internet indexing services (Kommadi, 2019).

Examples of Computer Systems and Data Structures

One example of a *Computer System Structure* is a computer's operating system. The operating system is a set of software programs that manage the computer's resources and provides a user interface for interacting with the computer. It also provides a platform for running other software applications (Glenn Brookshear & Brylow, 2020).

Some examples of well-known operating systems are Windows, UNIX, Mac OS, and Linux. Mobile devices, such as phones, tablets, other intelligent devices, and wearable devices, use specific operating systems that are only compatible with specific intelligent devices (My-course.co.uk, 2017). For example, iOS is always run on an iPhone, whereas Android is always run on a Google Pixel phone.

Operating systems make computers convenient and straightforward to use.

An example of a *Data Structure* is an array, a collection of elements of the same data type. An array typically stores data that needs to be accessed and manipulated

sequentially. Another example of a data structure is a linked list, a collection of elements linked together using pointers. Linked lists typically store data that needs to be accessed and manipulated randomly.

(Bullinaria, 2019).

Why are System and Data Structures Important?

System and data structures are important because they provide the foundation for building efficient and effective software applications. By understanding the underlying system and data structures, software developers can write programs optimised for the specific hardware and software environments they are working with.

For example, if software developers understand how data is stored in memory, they can write code optimised for accessing and manipulating that data. Similarly, if software developers understand how the operating system manages system resources, they can write code optimised for using those resources efficiently.

1. **Operating Systems**

An operating system (OS) is software that manages computer hardware resources and provides standard services for computer programs. The operating system is the most critical software in a computer system. It manages computer resources such as memory, input/output devices, and disk drives. An operating system also provides a user interface to interact with the computer, enabling application programs to run (www.tutorialspoint.com, N.D.).

Operating System Components

An operating system comprises several components that provide the required functionality. These components include the kernel, device drivers, system calls, shell, and file system.

Kernel: The kernel is the core component of an operating system. It provides low-level services to other parts of the operating system and user applications. The kernel manages memory, input/output, and process management.

Device Drivers: Device drivers are programs that allow the operating system to interact with hardware devices such as keyboards, printers, and network adapters.

System Calls: System calls interface user applications and the operating system. They allow applications to request services from the operating system.

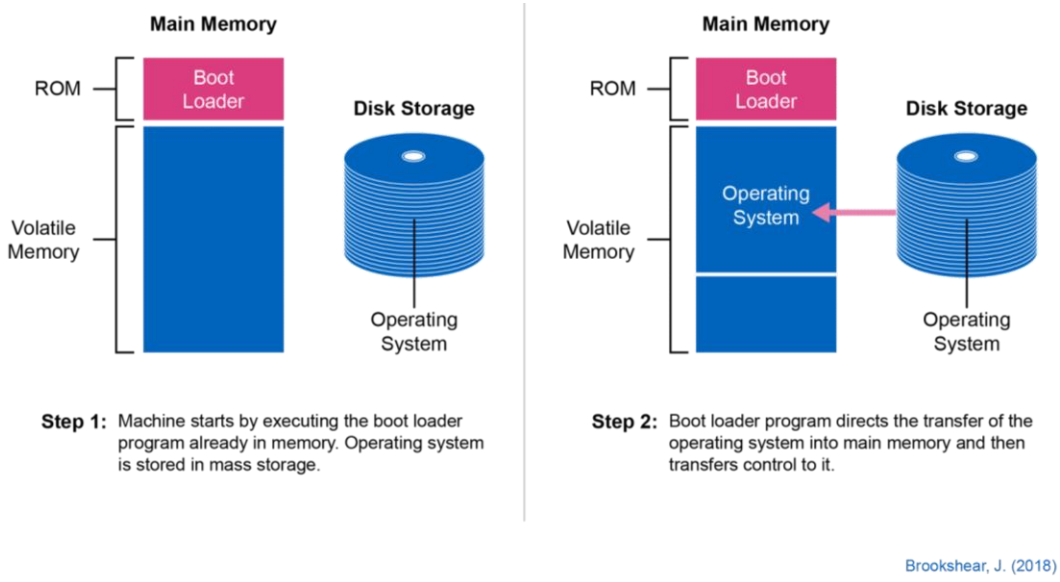
Shell: The shell program provides a command-line interface for interacting with the operating system. It allows users to execute commands and scripts.

File System: The file system manages to store and retrieve files on disk or other storage media. It provides a hierarchical organisation of files and directories.

(Miller, 2023).

Booting Process

The booting process is the process of starting a computer system from a powered-off or reset state. When the computer is turned on, the operating system is loaded into memory, and the hardware is initialised. The booting process comprises several steps: BIOS initialisation, bootloader loading, kernel loading, and initialisation.



Brookshear, J. (2018)

BIOS Initialization: The BIOS (Basic Input/Output System) is a program that initialises and checks the hardware components of the computer system. It loads the bootloader into memory.

Bootloader Loading: The bootloader program loads the operating system kernel into memory. It is typically located on a hard disk or other storage media.

Kernel Loading: The kernel is loaded into memory by the bootloader. It initialises the system hardware and prepares the system for user applications.

Initialisation: The operating system initialises and starts the necessary system services, device drivers, and other components required for user applications.

Operating System Threats

An operating system is vulnerable to various threats, including viruses, malware, and hackers. These threats can damage the system, steal sensitive information, and cause harm.

Viruses: A virus is a program that attaches itself to a legitimate program and infects it. The virus can replicate itself and spread to other programs and systems.

Malware: Malware is malicious software designed to harm the computer system or steal sensitive information. It includes spyware, adware, and ransomware.

Hackers: Hackers attempt to gain unauthorised access to a computer system. They can steal sensitive information, cause system damage, and disrupt system operations.

The following steps are taken to reduce this risk: maintaining a log (auditing), using utility programmes, and restricting rights and privileges.

(My-course.co.uk, 2017).

2. **Data Structures**

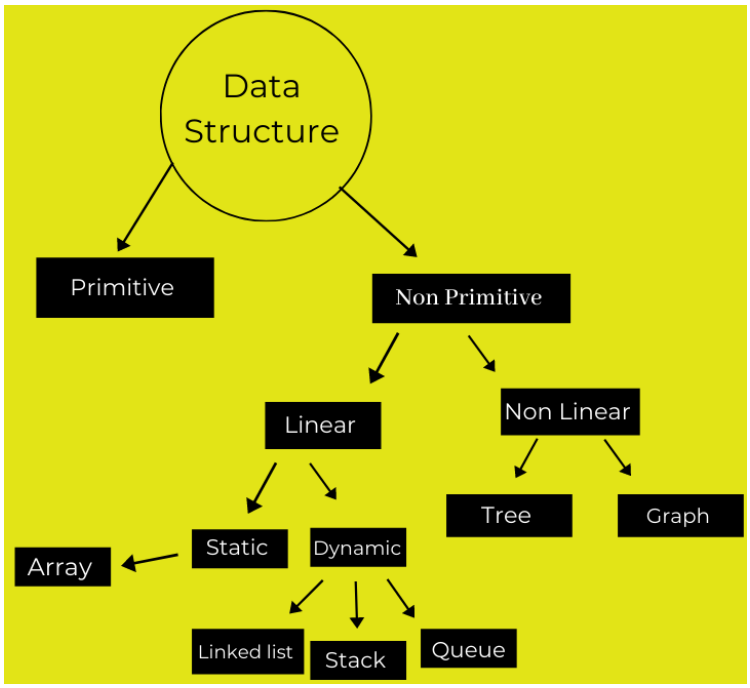
A data structure is used to organise data in memory. There are numerous methods for managing data in memory, such as arrays, lists, stacks, queues, etc. Furthermore, the data structure is a collection of algorithms that can be used to organise data in memory in any programming language (Thareja, 2014).

Abstract data types are the name given to these algorithms, and abstract data types are merely a set of rules.

(R, 2021).

Data abstraction is a programming concept that allows programmers to define and manipulate complex data types without exposing their implementation details. It simplifies the complexity of data structures and enables code reusability.

Data Structures are classified into two types: primitive data structures and non-primitive data structures.



(R, 2021).

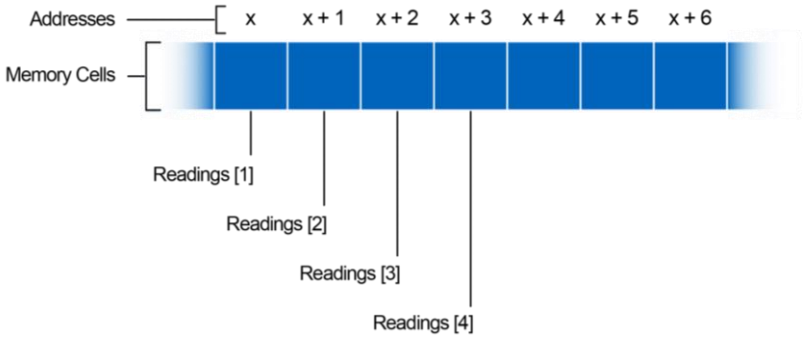
Primitive & Non-Primitive Data Types

Primitive Data Structures operate directly on machine instructions. These are the basic data types. Integer, char, float, double, and pointer data types are primitive data structures that can only hold one value.

Non-primitive Data Structures are those that are more complex than primitive data structures. Non-primitive data types are further classified into Linear Data Structure and Non–Linear Data Structure.

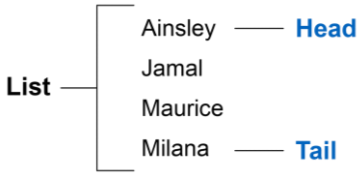
A linear data structure comprises data elements arranged sequentially, with each element connected to the elements before and after. This connection allows you to run through a linear arrangement simultaneously. Because memory is also sequential, such data structures are simple to implement. Lists, Queues, Stacks, and Arrays are examples of linear data structures (R, 2021).

An array is a collection of similar data elements stored in adjacent memory locations. The most basic data structure is one in which each data element can be accessed directly by its index number (R, 2021).



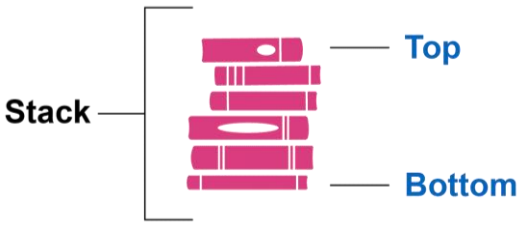
Brookshear, J. (2018)

A linked list is a linear data structure in computer memory that maintains a list-like structure. It is a collection of nodes not stored in the same place. Pointers connect each node of the list to its neighbouring node (R, 2021).



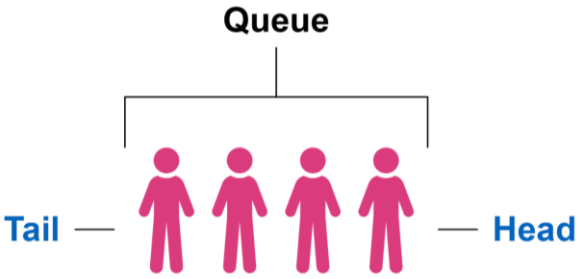
(Brookshear, 2018).

A stack is a linear data structure in which operations are performed in a specific order. The order could be FILO (First In, Last Out) or LIFO (Last In, First Out) (Last In First Out) (R, 2021).



(Brookshear, 2018).

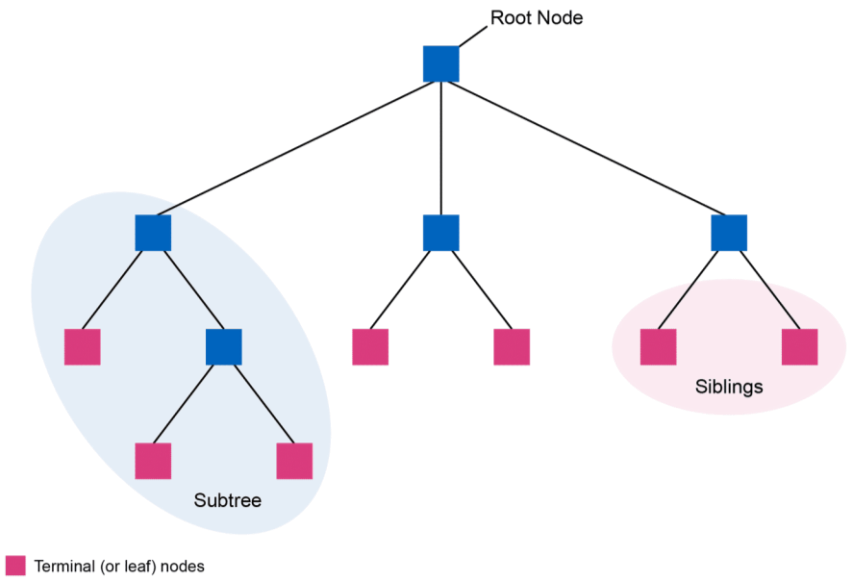
A queue is a linear data structure in which elements can be inserted from only one end (the rear) and deleted from the other (the front). It adheres to the FIFO (First In, First Out) principle (R, 2021).



(Brookshear, 2018).

Non-linear Data Structures have no fixed sequence for connecting all their elements, and each element can have multiple paths to connect to other elements. Such data structures can support multi-level storage and are not always traversable in a single run. Such data structures are difficult to implement but are more memory efficient. Examples include trees, BST, Graphs, and other non-linear data structures (R, 2021).

A tree is a multilevel data structure made up of nodes. The root node is at the top of the tree, while the leaf nodes are at the bottom. Each node has a single parent but multiple children (R, 2021).



(Brookshear, 2018).

A graph is a graphical representation of a collection of objects linked together by edges. The points that connect the interconnected nodes are called vertices, and the links that connect the vertices are called edges (R, 2021).

Classification of Data Structure

Static data structures and dynamic data structures are two types of data structures.

Static data structures are those whose size is determined at compile time. As a result, the maximum size is fixed and cannot be altered.

Dynamic data structures are data structures whose size is determined at run time. As a result, the maximum size is adaptable and can be changed as needed.

Operations on Data Structures

The following are some of the most common operations that can be performed on data structures: Searching, Sorting, Insertion, Deletion, and Updation.

Conclusion

In conclusion, systems and data structures are essential concepts in computer science that are important for understanding how software applications work. The system refers to components that perform a specific function or task. In contrast, data structure refers to the way in which data is organised and stored within a computer program or application. By understanding these concepts, software developers can write programs optimised for the specific hardware and software environments they work with, leading to more efficient and effective software applications.

Reference:

Uri.edu. (2019). *Introduction To Computers*. [online] Available at: https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading01.htm.

Kommadi, B. (2019). *Learn Data Structures and Algorithms with Golang: Level up your Go programming skills to develop faster and more efficient code*. Packt Publishing Ltd.

Glenn Brookshear, J. & Brylow, D. (2020). *Computer Science An Overview THIRTEENTH EDITION GLOBAL EDITION*. [online] Available at: https://pdfuni.com/sample/Science/SC301-400/SC830/sample-Computer%20Science%20An%20Overview%2013th%2013E%20Glenn%20Brookshear.pdf.

My-course.co.uk. (2017). *Operating Systems & Data Structures*. [online] Available at: https://www.my-course.co.uk/Computing/Computer%20Science/LCS/LCS%20Lecturecast%203/content/index.html#/.

Thareja, R. (2014). Data structures using C. Available at: http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/5374/1/Data%20structures%20using%20C%2C%202nd%20Ed.%20by%20Thareja%2C%20Reema%20%282014%29.pdf [Accessed 26 Mar. 2023].

Bullinaria, J. (2019). *Lecture Notes for Data Structures and Algorithms Revised each year*. [online] Available at: https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf.

TutorialsPoint (N.D). *Operating System - Overview - Tutorialspoint*. [online] Available at:

https://www.tutorialspoint.com/operating_system/os_overview.htm#:~:text=An%20Operating%20System%20(OS)%20is.

Miller, S. (2023). *What Is an Operating System?* [online] Codecademy Blog. Available at: https://www.codecademy.com/resources/blog/operating-system/ [Accessed 7 Apr. 2023].

R, V. (2021). *What is Data Structure: Need, Types & Classification*. [online] Great Learning Blog: Free Resources What Matters to shape your Career! Available at: https://www.mygreatlearning.com/blog/data-structure-tutorial-for-beginners/#:~:text=Data%20Structures%20are%20necessary%20for.