Module Code & Module Title

Level 5 – CT5052NP Network Operating Systems

Assessment Type

Logbook .

Semester

2023/24 Spring/Autumn

Student Name: Dishant Giri

London Met ID: 23048800

College ID: NP04CP4A230180

Submitted To: Prashant Adhikari

Word Count ():

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

1. Introduction to kernel

Kernel is a small piece of code that is loaded into memory when computer boots which contains instructions that manages hardware devices, memory allocation, system processes, and other programs.

Major components & functions of kernel:

i. Process Management

- Scheduling and execution of processes.
- Context switching between processes.
- Process creation and termination.

ii. **Memory management**

- Allocation and deallocation of memory space.
- Managing virtual memory.
- Handling memory protection and sharing.

iii. Device management

- Managing input/output devices.
- Providing a unified interface for hardware devices.
- Handling device driver communication.

iv. Resources Management

- Managing system resources like CPU time, disk space, network bandwidth, etc.
- Allocating and deallocating resources as needed.
- Monitoring resources usage and enforcing resource limits.

v. Security and Access Control

- Enforcing access control policies.
- Managing user permissions and authentication.
- Ensuring system security and integrity.

2. Types of kernel

There are many types of kernel but lets discuss about any two of them which are:

i. Type of kernel in Operating Systems

- **Monolithic Kernel**

  A monolithic kernel is a kernel architecture where the entire kernel is run in kernel space in supervisor mode. In common with other architectures (microkernel, hybrid kernels), the kernel defines a high-level virtual interface over computer hardware, with a set of primitives or system calls to implement

operating system services such as process management, concurrency, and memory management in one or more modules. (Ivan Stankov, Grisha Spasov, 2006)

- **MicroKernel**
  A microkernel only includes the most fundamental services like inter-process communication, basic scheduling, and memory management. Other services like drivers, file systems, and network protocols run in user space.

ii. Type of kernel in Machine learning

- **Linear Kernel**
  A **linear kernel** is a type of kernel function commonly used in machine learning, particularly with algorithms like Support Vector Machines (SVM). It is defined as the inner (dot) product of two vectors in the input feature space.
- **Radial Basis Function (RBF) Kernel**
  The RBF kernel, also known as the Gaussian kernel, maps the data into an infinite-dimensional space. It's typically used when the data is not linearly separable.

3. Popular kernels and their history

Kernels used by popular operating systems like Windows, IOS and Ubuntu are as follows:

i. IOS(Darwin Kernel)

iOS uses the XNU kernel, which stands for "X is Not Unix." It is part of the **Darwin** operating system, a hybrid kernel that combines elements from both monolithic and microkernel architectures.

The Mach kernel project was started in 1985 at the Carnegie Mellon University in Pittsburgh. Its goal was to build a true micro kernel which could act as a replacement for the BSD kernel. At the time the generally accepted theory was that micro kernels would eventually replace the monolithic kernels used at the time. In a micro kernel design as much code as possible is moved from kernel space to user space in so called services, making the kernel a message parsing service between the different services and between the hardware and its drivers in user space. This design should improve the overall stability since crashed user space services could be restarted without panicking the kernel, as well as offer a better security model by running services, such as drivers, under a less privileged account. While the design of a micro kernel is favorable, there were considerable perfor mance issues. The first versions of the Mach kernel still contained the entire BSD kernel and was thus far from being micro. The idea was to slowly phase out the BSD kernel components and move them to user space, in separated services. Since the different services should be able to communicate with each

other, Mach implemented a Inter Process Communication (or IPC) system, which allowed user space applications to send messages to each other. However, because for each message multiple context switches were needed (from user space to kernel space and back to user space) and the access rights had to be checked, perfor mance were poor. For instance, a dummy syscall on BSD would take 40 µs where its Mach counterpart would take nearly 500 µs[20]. Due to these performance problems the project was canceled in 1994. (Keuper, 2012)

ii. Windows(NT Kernel)

Windows uses the NT kernel, which is also a hybrid kernel. It combines aspects of microkernel and monolithic design to balance performance and modularity.

When Microsoft decided to build a successor to MS-DOS, it was strongly influenced by the success of the Macintosh. It produced a GUI-based system called Windows, which originally ran on top of MS-DOS (i.e., it was more like a shell than a true operating system). For about 10 years, from 1985 to 1995, Windows was just a graphical environment on top of MS-DOS. However, starting in 1995 a freestanding version of Windows, Windows 95, was released that incorporated many operating system features into it, using the underlying MS-DOS system only for booting and running old MS-DOS programs. In 1998, a slightly modified version of this system, called Windows 98 was released. Nevertheless, both Windows 95 and Windows 98 still contained a large amount of 16-bit Intel assembly language. Another Microsoft operating system is Windows NT (NT stands for New Technology), which is compatible with Windows 95 at a certain level, but a complete rewrite from scratch internally. It is a full 32-bit system. The lead designer for Windows NT was David Cutler, who was also one of the designers of the VAX VMS operating system, so some ideas from VMS are present in NT. In fact, so many ideas from VMS were present in it that the owner of VMS, DEC, sued Microsoft. The case was settled out of court for an amount of money requiring many digits to express. Microsoft expected that the first version of NT would kill off MS-DOS and all other versions of Windows since it was a vastly superior system, but it fizzled. Only with Windows NT 4.0 did it finally catch on in a big way, especially on corporate networks. Version 5 of Windows NT was renamed Windows 2000 in early 1999. It was intended to be the successor to both Windows 98 and Windows NT 4.0. (Tanenbaum, 2009)

iii. Ubuntu(Linux Kernel)

Linux is a phenomenon of the Internet. Born out of the hobby project of a student it has grown to become more popular than any other freely available operating system. To many Linux is an enigma. How can something that is free be worthwhile? In a world dominated by a handful of large software

> corporations, how can something that has been written by a bunch of ``hackers" (sic) hope to compete? How can software contributed to by many different people in many different countries around the world have a hope of being stable and effective? Yet stable and effective it is and compete it does. Many Universities and research establishments use it for their everyday computing needs. People are running it on their home PCs and I would wager that most companies are using it somewhere even if they do not always realize that they do. Linux is used to browse the web, host web sites, write theses, send mail and, as always with computers, to play games. Linux is emphatically not a toy; it is a fully developed and professionally written operating system used by enthusiasts all over the world. (Rusling, 2001)

4. Boot process

The **boot process** is the sequence of operations that a computer system undergoes when it is powered on, leading to the loading of the operating system (OS). The boot process is crucial because it initializes the hardware and loads the kernel, allowing the OS to take control of the system.

As briefly mentioned before, the parentboard, also known as the motherboard, contains the BIOS (Basic Input/Output System), which is where the Pentium boot process starts. The flash RAM-based BIOS first checks the RAM and other fundamental devices, such as the keyboard. It looks for plug-and-play and legacy devices on the ISA and PCI busses. If modifications are found, the device configuration is changed.

The boot device is then selected by the BIOS, which normally tries the hard drive, CD-ROM, and floppy disk in order. The software to check the partition table, load the secondary bootloader, and load the operating system is included in the first sector of the boot device that is read into memory.

After the OS has loaded, it loads the required device drivers and asks the BIOS for configuration information. The OS launches a login prompt or a graphical user interface after starting background tasks.

This procedure makes ensuring that all hardware has been correctly setup and software has been loaded, preparing the system for user interaction. (Tanenbaum, Modern Operating Systems, 2015)

5. summary

This report looks at the kernel, which is main part of an operating system that helps manage hardware and resources. We discussed it’s key functions like managing processes, memory, devices, and security.

We also covered two types of kernels: the monolithic kernel, which runs everything together, and the microkernel, which keeps only essential services in the kernel and moves others to user space.

We highlighted popular kernels like the XNU kernel used in ios, the NT kernel in windows, and the linux kernel in Ubuntu, each with its own background.

Finally, we explained the boot process, which gets the hardware ready and loads the operating system, so everything works for the user. Understanding these basics helps us see how operating systems function.

6. Bibliography

Ivan Stankov, Grisha Spasov. (2006). *Discussionof Microkernel and Monolithic Kernel Approaches*.

Keuper, D. (2012). *XNU: a security evaluation*.

Rusling, D. A. (2001). *The linux kernel*.

Tanenbaum, A. S. (2009). *modern operating system.*

Tanenbaum, A. S. (2015). *Modern Operating Systems.*