Nanyang Polytechnic

IT2164/IT2561 Operating Systems and Administration

Chapter 11

OS Design

Objectives

- After this lesson, you will be able to :
  - Understand the different considerations associated with OS design.
  - Know the essential characteristics of monolithic kernels.
  - Know the design of an object-oriented OS.
  - Know the design of Linux and Windows NT kernels.

Design Considerations

- An operating system is a large collection of software with complex input/output relationships. Hence, designing an OS is subjected to many considerations
  - Performance
  - Trusted Software
  - Modularization
  - Portability

Performance

- Although OS is essential in a computer system, OS activity is an overhead.
- Even though CPU and hardware are getting more powerful and cheaper, performance is still top priority in OS design.
- Complex OS design requires structured programming techniques, such as OO.
- However, fast code is usually not achieved by OO languages, but structured programming. E.g., C versus C++.

Trusted Software

- Expanded use of computers result in requirements for trustworthy software.
- OS kernel should be designed to ensure sensitive operations can only be executed by the OS in supervisor mode. Misbehaving programs are stopped before they wreak havoc on the computer system.

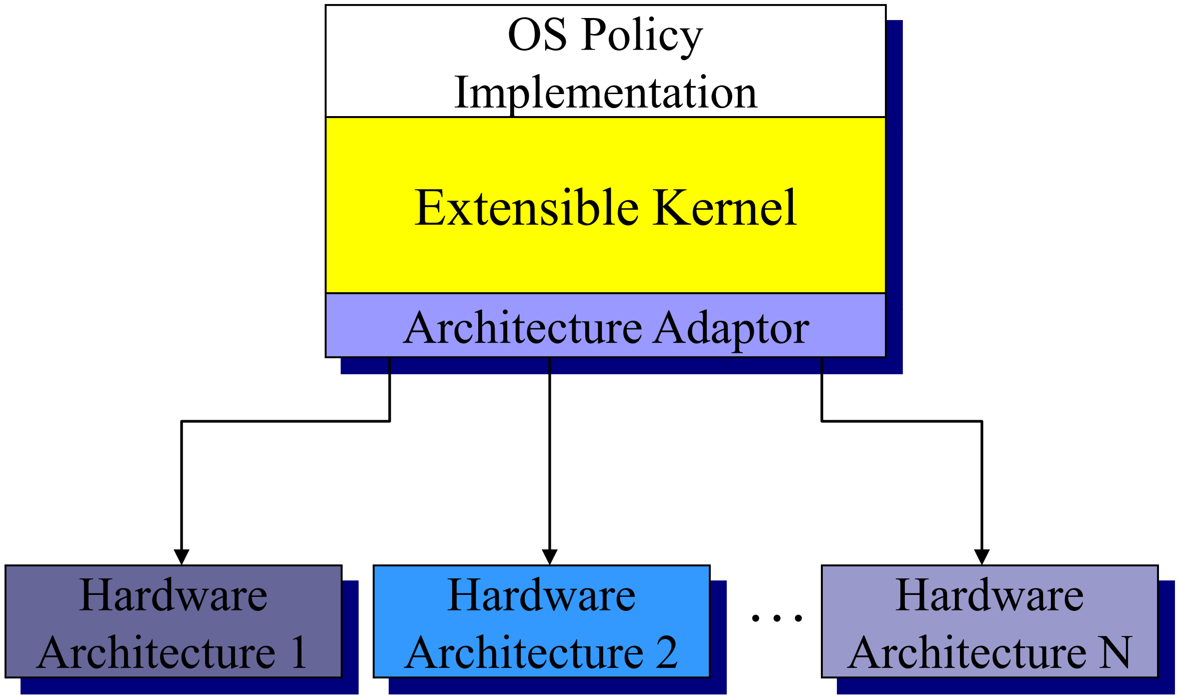
Modularization

- Different portions of the OS should be implemented as independent software modules.
- This allows for one module to be modified without affecting other areas of the OS.
- Allows for customizing the interfaces of the OS for peripheral devices without the need to recompile the whole kernel. (e.g., Linux)

Portability

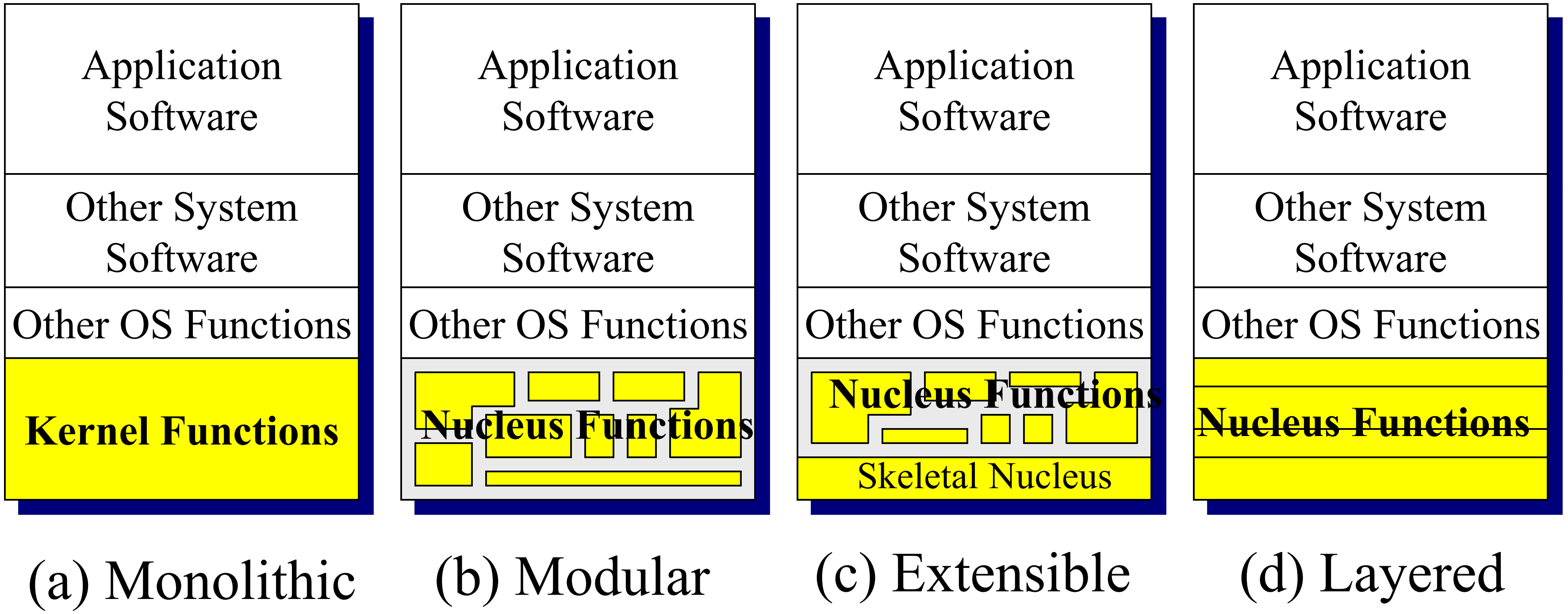
- Refers to the ability to run on different hardware platforms.
- Modern OS are designed with separate modules that takes care of the abstraction of hardware functions.
- This enables a OS to be ported to another platform just by re-writing the portion that is responsible for the abstraction.
- In Windows, the hardware abstraction layer (HAL) is the low-level software module responsible for the abstraction.

Porting an OS to Different Architectures



- Used in Windows NT Technology
- Used in Linux (sort of …)

OS Software Organization



Monolithic Kernels

- Windows and UNIX are monolithic kernels by design.
- All software and data structures are placed in one logical module.
- Efficient if well implemented.
- Can be difficult to understand and maintain.
- Being in one logical module, different portions of the OS can access one another directly, leading to performance.

MS-DOS as a Monolithic Kernel

- OS kernel implemented in the read-only Basic Input/Output System (BIOS) routines and 2 executable files named IO.SYS and MSDOS.SYS.
- Two files CONFIG.SYS and AUTOEXEC.BAT allows for OS extensions (drivers), that allows the kernel to interface with additional hardware.

UNIX as a Monolithic Kernel

- Introduced by researchers in AT&T.
- Traditional design implemented as a single logical module.
- Process and resource management, memory management and file management are implemented within a single software module.

Object Oriented OS

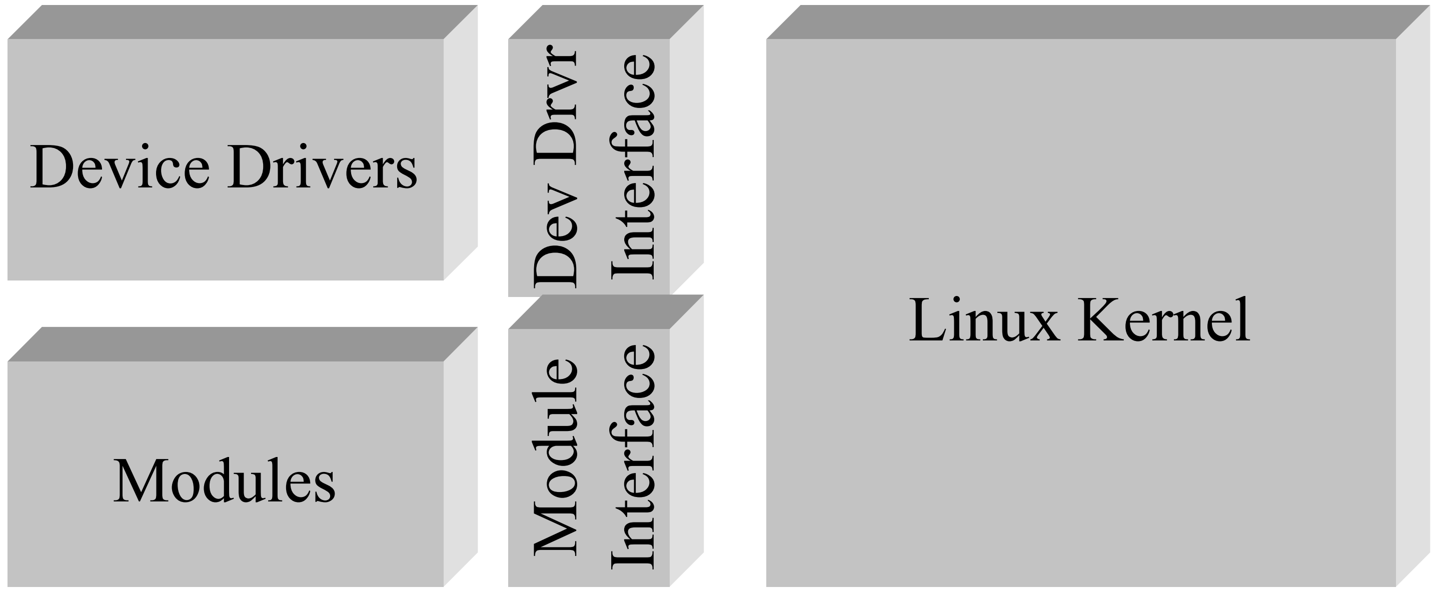
- A modular kernel which is partitioned among logically independent components.
- Each component needs to interact with one another through interfaces.
- Easier to maintain and modify as borders are clearly defined.
- Use of interfaces lead to inefficiencies.
- No commercial implement as yet.

Nanyang Polytechnic

Linux Kernel

- Based on UNIX design, and so is a monolithic kernel.
- Implements modules as a mechanism to add new code to the OS. E.g., for device drivers.

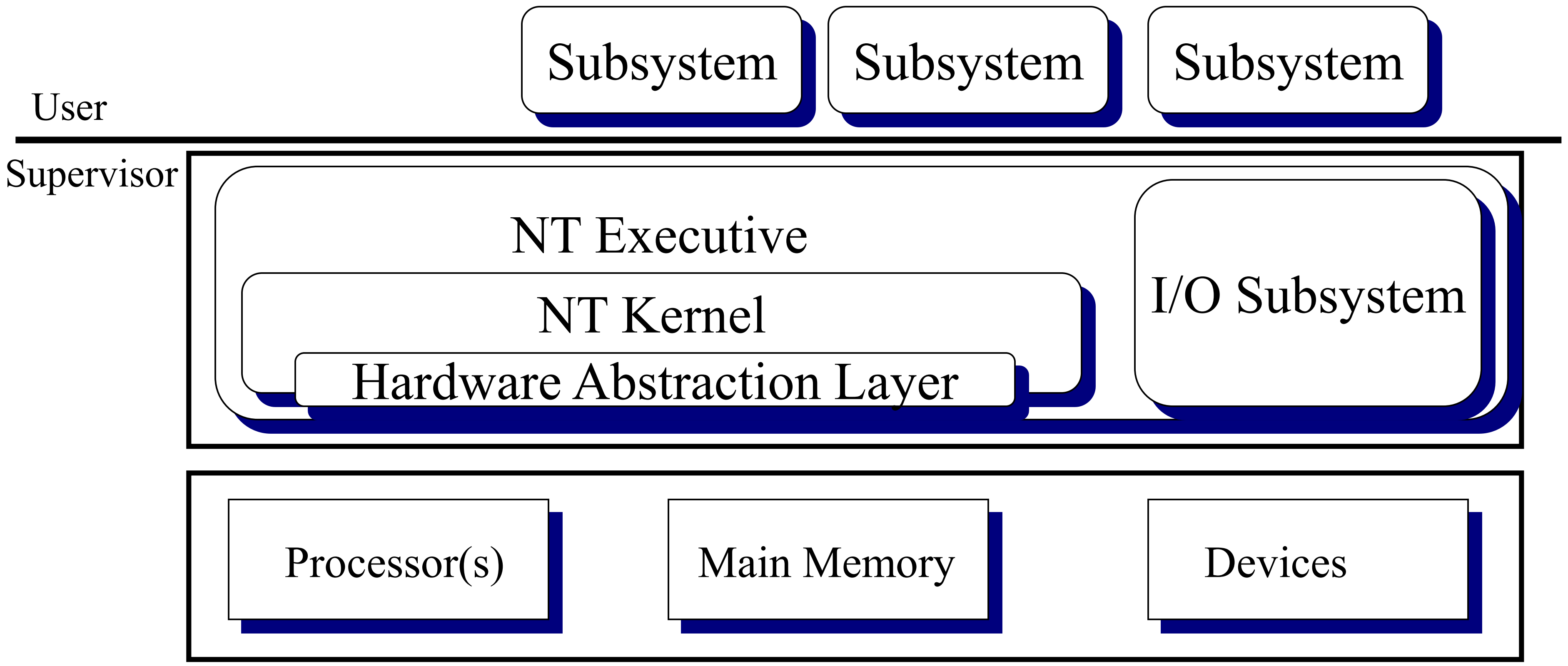
Linux Kernel Organization



Windows Kernel

- Monolithic by design.
- Kernel provides the objects and threads on the HAL and hardware layers.
- NT Executive builds on the NT Kernel to implement the Windows policies and services.
- Includes
  - Object Manager
  - Process & Thread Manager
  - Virtual Memory Manager
  - I/O Manager
  - Cache Manager

Windows NT/2000 Organization



Conclusion

- Modern OS design is based on years of evolution.
- Although commercial OSs are based on the monolithic design to achieve performance, work is being done on other better design paradigms to further improve future OS.