

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Elective Course 4

ELC-467 Operating Systems

Dr. Hany M. Elsayed

e-mail: helsayed@ieee.org

Course material available at: <http://eece.cu.edu.eg/moodle/>

References

- A. Silberschatz, G. Gagne and P.B. Galvin , *Operating System Concepts* , 10th Ed., John Wiley, 2018.
- A.S. Tanenbaum, *Modern Operating Systems*, 4th Ed., Pearson, 2014.
- W. Stallings, *Operating Systems: Internals and Design Principles*, 9th Ed., Pearson, 2017.

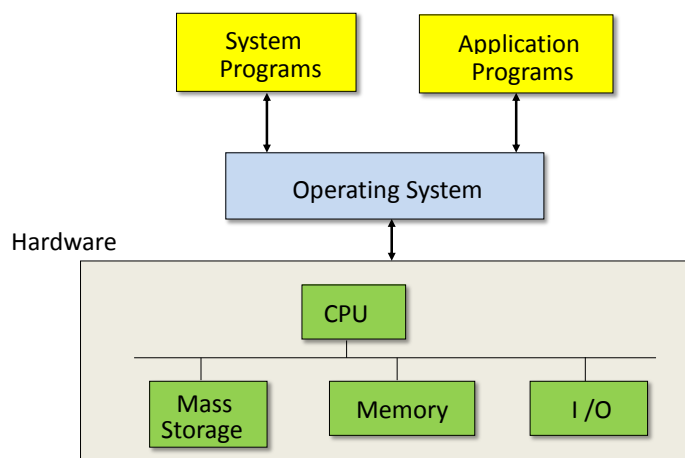
What is an Operating System?

An Operating System (OS) is a set of programs that:

- Manages the computer hardware resources efficiently.
- and
- Provides a convenient user interface to the computer.

The operating system has a major influence on the overall function and performance of any computing system.

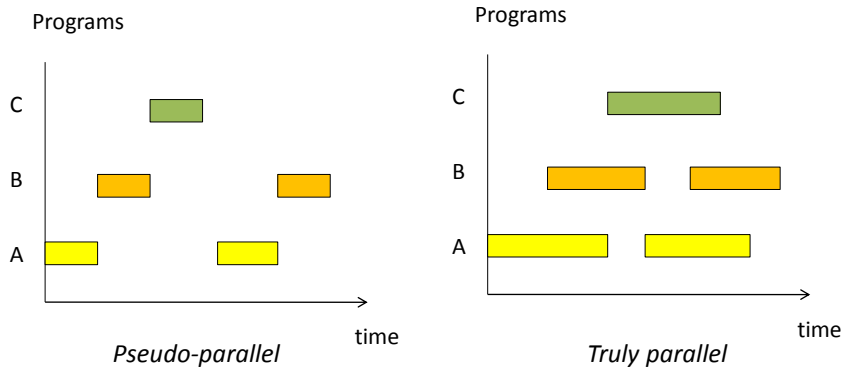
What is an Operating System?



Types of Operating Systems

□ Single Programmed vs. Multitasking Systems

How many programs are active at the same time?



ELC 467– Spring 2020

Lecture 1- Page 3

Types of Operating Systems

□ Single Processor vs. Multiple Processor Systems

- Single processor systems.
- Tightly coupled multiple processor systems.
With common memory and short communication delays.
(Symmetric or heterogeneous multiprocessors, Multicore systems.)
- Loosely coupled multiple processor systems.
Autonomous nodes connected by a network. No common memory and relatively long communication delays.
(Clustered systems, Distributed systems).

ELC 467– Spring 2020

Lecture 1- Page 4

Types of Operating Systems

❑ Embedded Systems vs. General-Purpose Systems

In an embedded system the computer is a part of a larger non-computing system. Typically system is dedicated to a particular set of tasks with critical performance requirements. In many cases, user have no direct access to the computer.

Embedded systems have cost, power, and performance constraints. Since requirements of applications vary widely, a high degree of configurability is needed in an embedded OS.

Types of Operating Systems

❑ Real-Time vs. Non-Real-Time Systems

Real-time operating systems are used to run applications with timing constraints. In a real-time system, correctness depends not only on the logical results of computation but also on the time at which these results are produced.

In systems with no real-time constraints, we are rather interested in the average response time and throughput of the system.

The Functions of an Operating System

For study purposes, the functions of an operating system are usually classified into functions of:

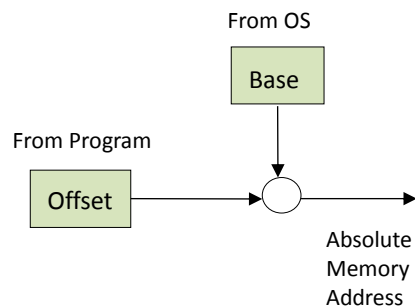
- ❑ Process Management
- ❑ Memory Management
- ❑ I/O Management
- ❑ File System Management

In actual implementation, each of these functions is not performed separately.

Some Basic Requirements for Multitasking

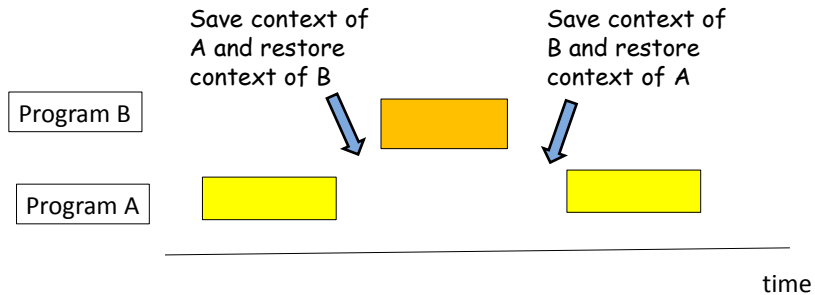
For efficient operation, a multitasking system should keep several programs in memory and switch CPU between them. All programs must thus be *relocatable*: i.e. can run from any available space in memory.

Thus, programs must use relative rather than absolute addresses.



Some Basic Requirements for Multitasking

While running, each program will have a *context*. This refers to all the information needed for its operation: register contents, memory pointers, ...etc. Multitasking requires continuous switching between program contexts.



Kernel Mode vs. User Mode

For proper operation, user programs should not be allowed to perform some operations:

- e.g. access memory of other programs.
- stop program switching.
- halt the processor.

Most advanced processors have two modes of operation:

Kernel mode: all instructions can be executed, used by OS.

User mode: some instructions are prohibited.

Several privilege levels may be available, e.g. Intel processors have four levels and ARMv8 processors have seven levels.