Welcome to the Webinar Offered by:

National Microelectronics Security Training (MEST) Center

Sponsors:

Introduction to PetaLinux

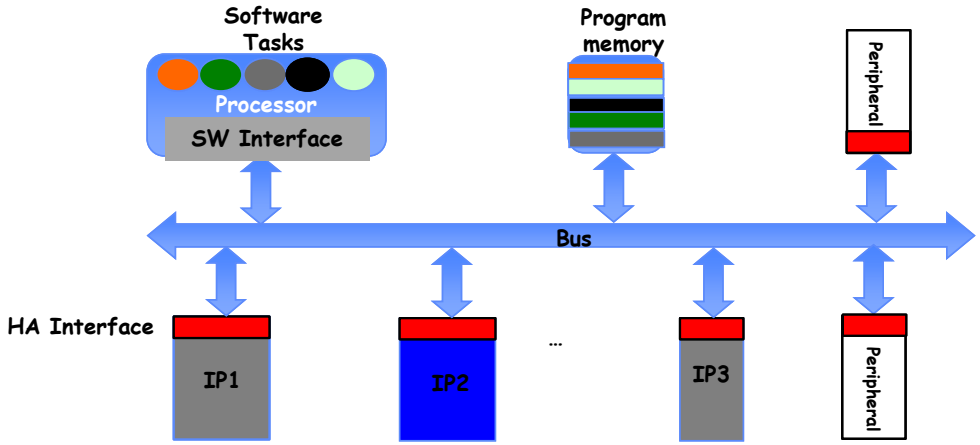
Embedded Operating Systems

Abigail Butka, Kawser Ahmed Muhammed, Wade Fortney,
Christophe Bobda

Agenda

- System-on-Chip Refresher
- Operating Systems
  - Definition
  - Uses
  - Benefits
  - Application to Systems-on-Chip
- Basic Lab Architecture
- Conclusion and Discussion

SoC- Architecture



Operating Systems

What is an Operating System?

- When you think of an operating system, what do you think of?

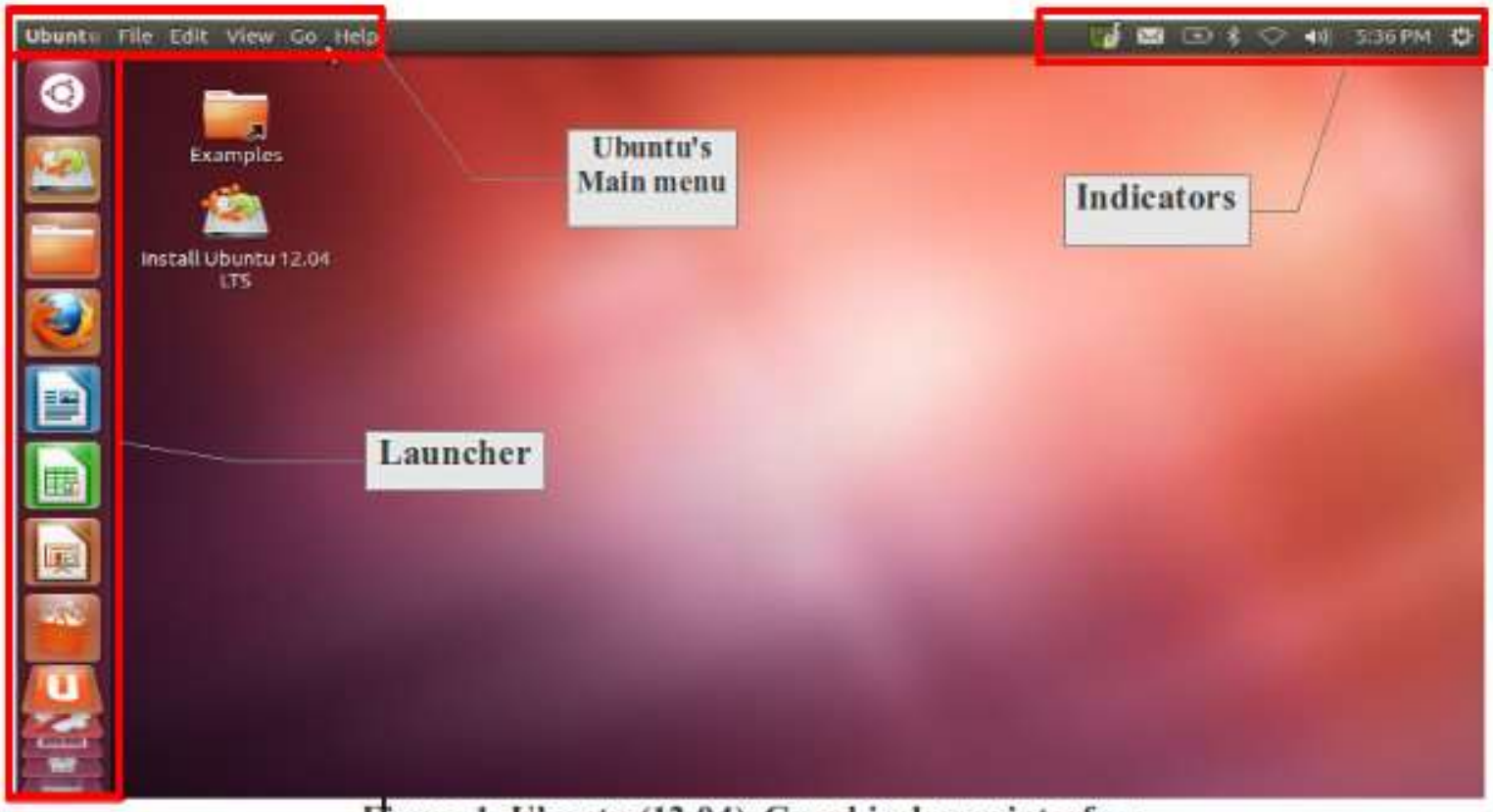


Figure 1. Ubuntu (12.04), Graphical user interface

What is an Operating System?

- An operating system is a layer of software that acts as an intermediary between the computer hardware and software applications

- Operating systems allow users to utilize hardware, like your computer's CPU, without requiring knowledge of how the hardware works

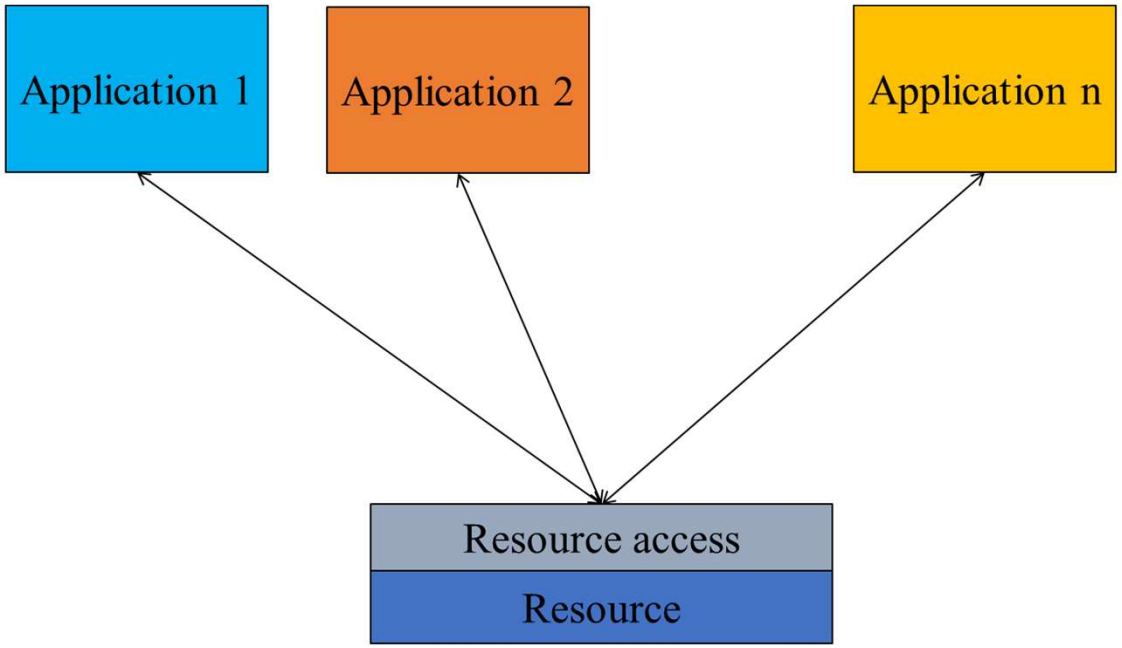
- Two main views to characterize operating systems
  - Service Provider
  - Resource Manager

Operating System as a Service Provider

- As a service provider, the operating system hides the low-level details from the user to provide a simplified experience

- Task
  - Application Programming Interfaces (APIs) to allow applications to request resources
  - User Interface or Graphical User Interface
  - File-system management

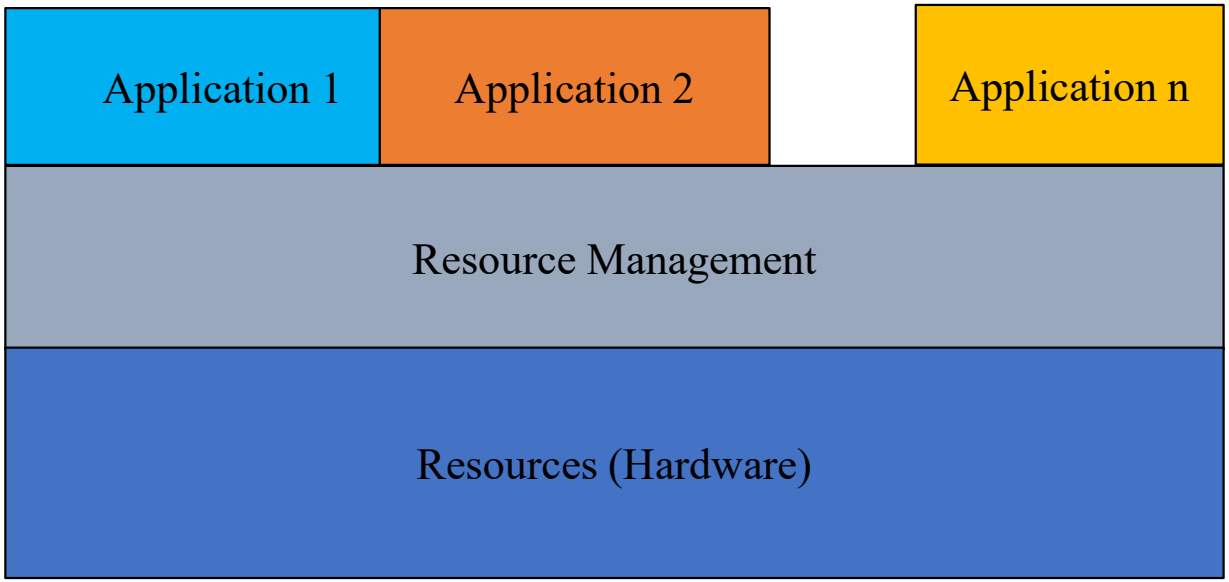
- Advantage
  - Performance
  - Modularity
  - Memory usage
  - Time to market
  - Reliability
  - Maintainability
  - Portability
  - Scalability



Operating System as a Resource Manager

- As a resource manager, the operating system ensures that each program gets its required time with and space on their required resources

- Task
  - Provide the environment under which program can run
  - Control and manage the access to resources by concurrent programs
  - Processor management
  - Memory management
  - Provide protection and security



Operating System Concepts

- The operating system is a layer of software that acts as an intermediary between the computer hardware and software applications. It provides:

- Process management
  - Processes need resources to accomplish their task
  - Upon completion, resources must be re-allocated
  - Can be single-threaded (sequential instructions) or multi-threaded (multiple instructions running at once)

- Memory management
  - Ensure all data/instructions are in memory before process execution
  - Determine what is where inside of the memory and when
  - Allocating and deallocating memory

- Operating system is responsible for:
  - Creating and deleting user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization, communication, and deadlock handling
  - File-system management
  - Etc…

Operating System Concepts

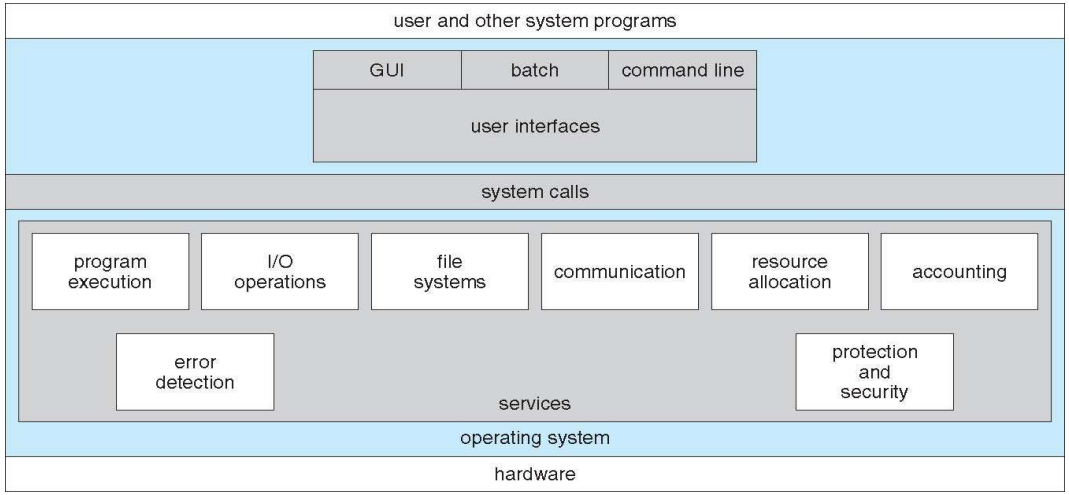
- Additionally, the operating system must provide protection and security

- Protection – any mechanism for controlling access of processes or users to resources defined by the OS
- Security – defense of the system against internal and external attacks

- Systems generally first distinguish among users to determine who can do what
- User identities (user IDs, security IDs) include name and associated number, one per user
- User ID then associated with all files and processes of that user to determine access control

Why Should we use Operating Systems?

- Operating systems provide many useful functions such as:
  - User interface, Graphics User Interface (GUI), and Command Line Interface
  - Program execution
  - I/O operations
  - File-system manipulation
  - Error detection
  - **Hardware resource management, mapping, and allocation**



How do we use Operating Systems in SoCs?

- PetaLinux
  - PetaLinux is a set of tools that can create a specialized, embedded Linux environment for Xilinx FPGAs
  - This Linux environment, or operating system, is commonly referred to as PetaLinux
  - The PetaLinux tools provide a workflow for creating, building, and deploying Linux

  - This operating system is unique because it is designed to work with FPGAs and the Vivado workflow
    - PetaLinux can import System-on-Chip .xsa designs
    - This allows the user to access and modulate GPIO pins or custom hardware from the operating system
    - Provides a high abstraction method of accessing low-level components

- PetaLinux creates an operating system similar to other Linux-based operating systems, only it can interface directly with Xilinx's hardware

Case Study and Tutorial