COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

EMBEDDED SYSTEM DESIGN

Multitasking - RTOS

Doan Duy, Ph. D.

Email: duyd@uit.edu.vn

COMPUTER ENGINEERING

Objective and Content

- Multitasking
- RTOS
- Verification and Validation

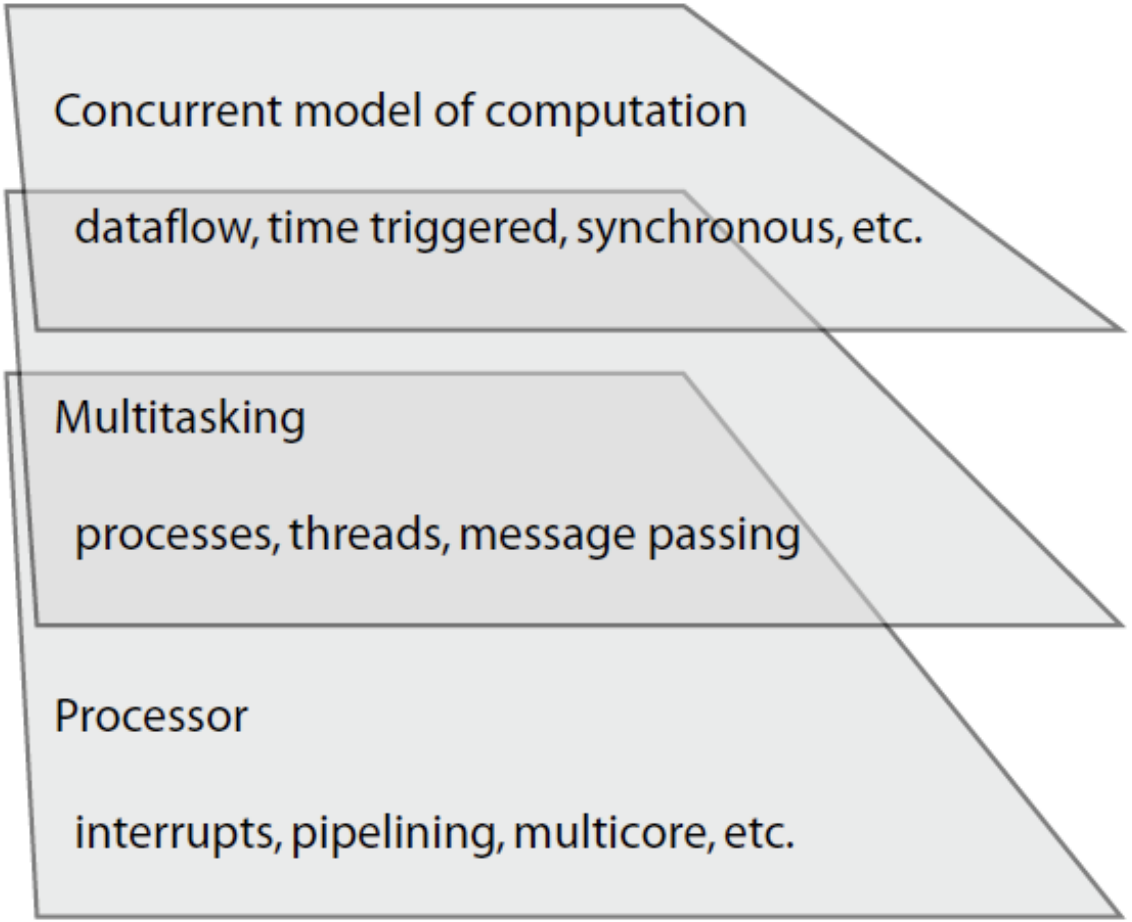
COMPUTER ENGINEERING

Objective and Content

- **Multitasking**
- RTOS
- Verification and Validation

COMPUTER ENGINEERING

Layers of Abstraction for Concurrency in Programs



Concurrency and Parallelism

- A program is said to be **concurrent** if different parts of the program *conceptually execute simultaneously*.

- A program is said to be **parallel** if different parts of the program *physically execute simultaneously* on distinct hardware.

A parallel program is concurrent, but a concurrent program need not be parallel.

COMPUTER ENGINEERING

Definition and uses

- Threads are sequential procedures that share memory.

- Uses of concurrency:
  - Reacting to external events (interrupts)
  - Exception handling (software interrupts)
  - Creating the illusion of simultaneously running different programs (multitasking)
  - Exploiting parallelism in the hardware (e.g. multicore machines).
  - Dealing with real-time constraints.

COMPUTER ENGINEERING

Multitasking - Threading

- Without an OS, multithreading is achieved with interrupts. Timing is determined by external events.
- Thread libraries (like “pthreads”): create, execute, schedule, suspense, block, terminate…
- Task scheduling: priorities, preemption policies, deadlines…
- Processes are collections of threads with their own memory, not visible to other processes.
- Segmentation faults are attempts to access memory not allocated to the process.
- Communication between processes must occur via OS facilities (like pipes or files).

Posix Threads (PThreads)

- PThreads is an API (Application Program Interface) implemented by many operating systems, both real-time and not. It is a library of C procedures.

- Standardized by the IEEE in 1988 to unify variants of Unix. Subsequently implemented in most other operating systems.

- An alternative is Java, which may use PThreads under the hood, but provides thread constructs as part of the programming language.

Creating and Destroying Threads

```
#include <pthread.h>

void* threadFunction(void* arg) {
    ...
    return pointerToSomething or NULL;
}

int main(void) {
    pthread_t threadID;
    void* exitStatus;
    int value = something;
    pthread_create(&threadID, NULL, threadFunction, &value);
    ...
    pthread_join(threadID, &exitStatus);
    return 0;
}
```

Can pass in pointers to shared variables.

Can return pointer to something.
Do not return a pointer to an local variable!

Create a thread (may or may not start running!)

Becomes arg parameter to threadFunction.
Why is it OK that this is a local variable?

Return only after all threads have terminated.

COMPUTER ENGINEERING

Issues of Multitasking

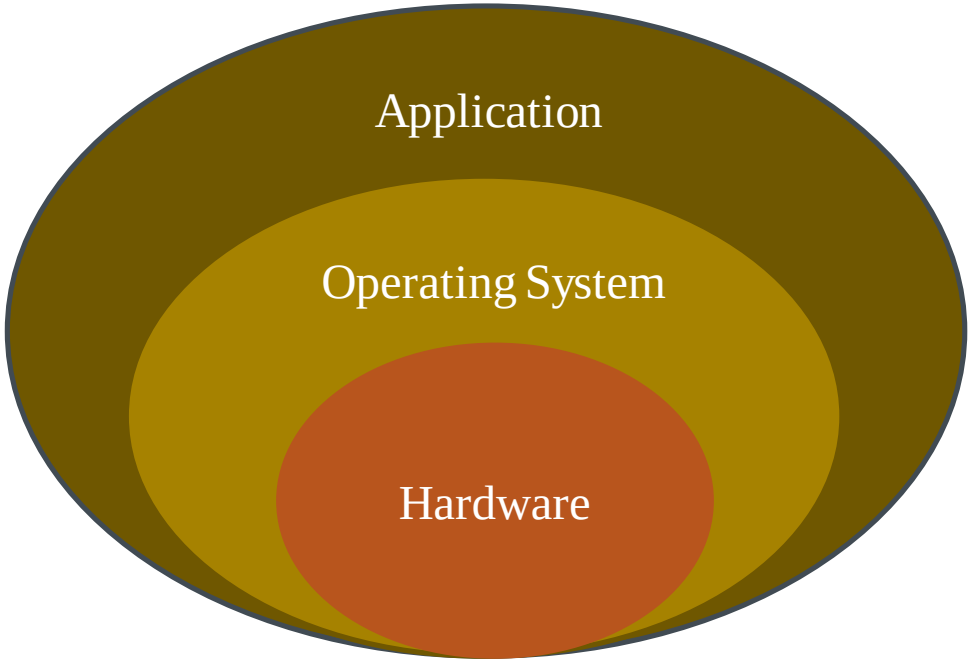
- Process/Thread management
- Memory management/sharing
- Task scheduling
- Process synchronization
- Deadlock control
- Fault Tolerance

COMPUTER ENGINEERING

Objective and Content

- Multitasking
- **RTOS**
- Verification and Validation

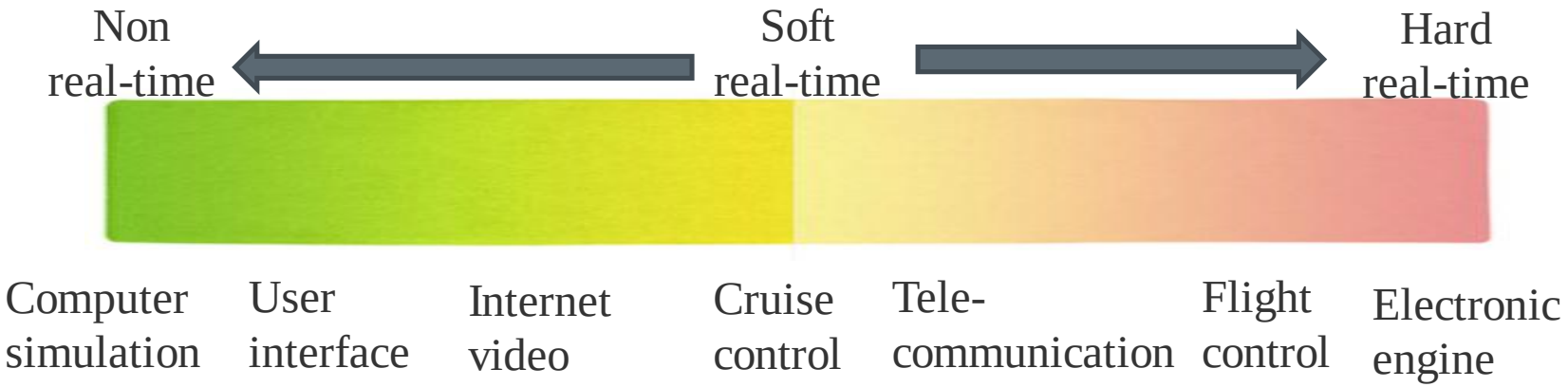
OS in Embedded Systems



- Hệ điều hành nhúng (embedded OS)
- Sensor-Node Operating System
- Smart-Card Operating System
- **Hệ điều hành thời gian thực (RTOS)**

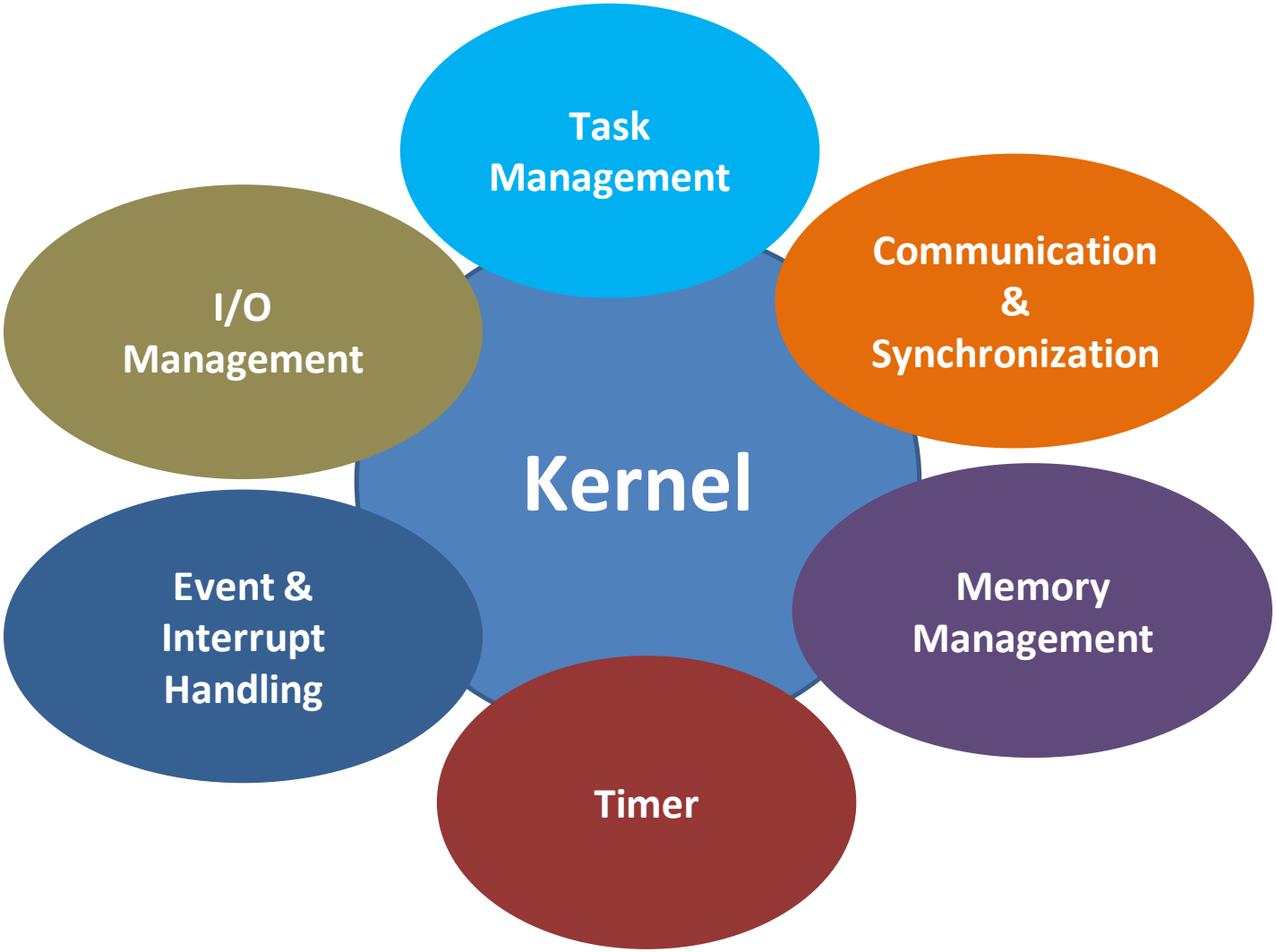
Real-time Operating System - RTOS

- RTOS là hệ điều hành được thiết kế cho các hệ thống thời gian thực.
- Yêu cầu cao về thời gian đáp ứng của hệ thống.
- Ví dụ hệ thống thời gian thực:



COMPUTER ENGINEERING

Basic components of RTOS



COMPUTER ENGINEERING

Characteristic of RTOS

Đa tác vụ
- Độ ưu tiên tác vụ
- Bộ định thời

Quản lý tài nguyên
- Bộ nhớ (mutex, semaphore…)
- Tài nguyên chia sẻ

RTOS

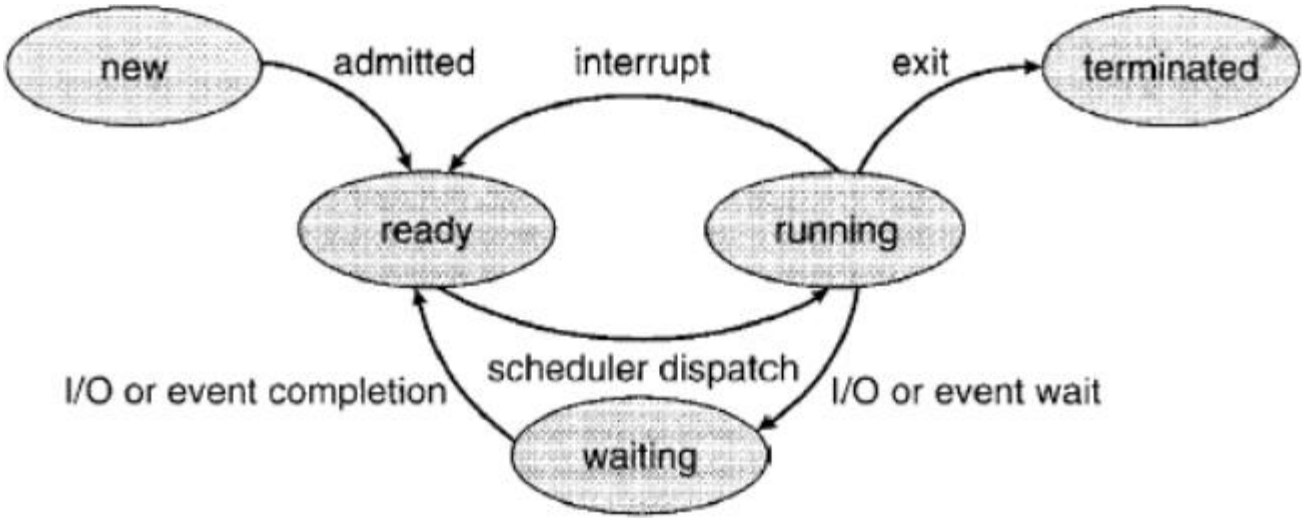
Tham số thời gian
- Thời gian hệ thống (tick)
- Thời gian xử lý ngắt
- Thời gian chuyển đổi ngữ cảnh

Tính tất định
- Deadline
- Trạng thái quá tải (overloaded)

COMPUTER ENGINEERING

Process in RTOS

- Phức tạp hơn các hạt nhân thực hiện theo cơ chế thăm dò và điều khiển ngắt
- Truyền tín hiệu logic bên trong các quá trình và các dịch vụ ngắt được tích hợp và thực hiện thông qua việc truyền dữ liệu
- Mô hình trạng thái của quá trình



COMPUTER ENGINEERING

Example of RTOS

- Features of FreeRTOS:

  - Pre-emptive or co-operative operation
  - Very flexible task priority assignment
  - Flexible, fast and light weight task notification mechanism
  - Queues
  - Binary semaphores
  - Counting semaphores
  - Mutexes
  - Recursive Mutexes
  - Software timers
  - Event groups
  - Tick hook functions
  - Idle hook functions
  - Stack overflow checking
  - Trace recording
  - Task run-time statistics gathering

  - Optional commercial licensing and support
  - Full interrupt nesting model (for some architectures)
  - A tick-less capability for extreme low power applications
  - Software managed interrupt stack when appropriate (this can help save RAM)

COMPUTER ENGINEERING

Objective and Content

- Multitasking
- RTOS
- **Verification and Validation**

Verification and Validation/Testing

- Both Verification and Validation are essential and balancing to each other.
- Different error filters are provided by each of them.
- Both are used to **find a defect** in different ways.

COMPUTER ENGINEERING

Deferences of Verification and Validation

Verification	Validation
Are we building the system right?	Are we building the right system?
Definition: Evaluating products of a development phase to meet the specified requirements.	***Definition:*** Evaluating products at the end of the development process to meet the customer expectations and requirements.
Objective: guarantee the requirements and specifications.	***Objective:*** guarantee the user's requirements
Activities: Reviews, Meetings and Inspections.	***Activities:*** Testing like black box testing, white box testing, scenario test.
In charge: QA team	***In charge:*** Testing team.

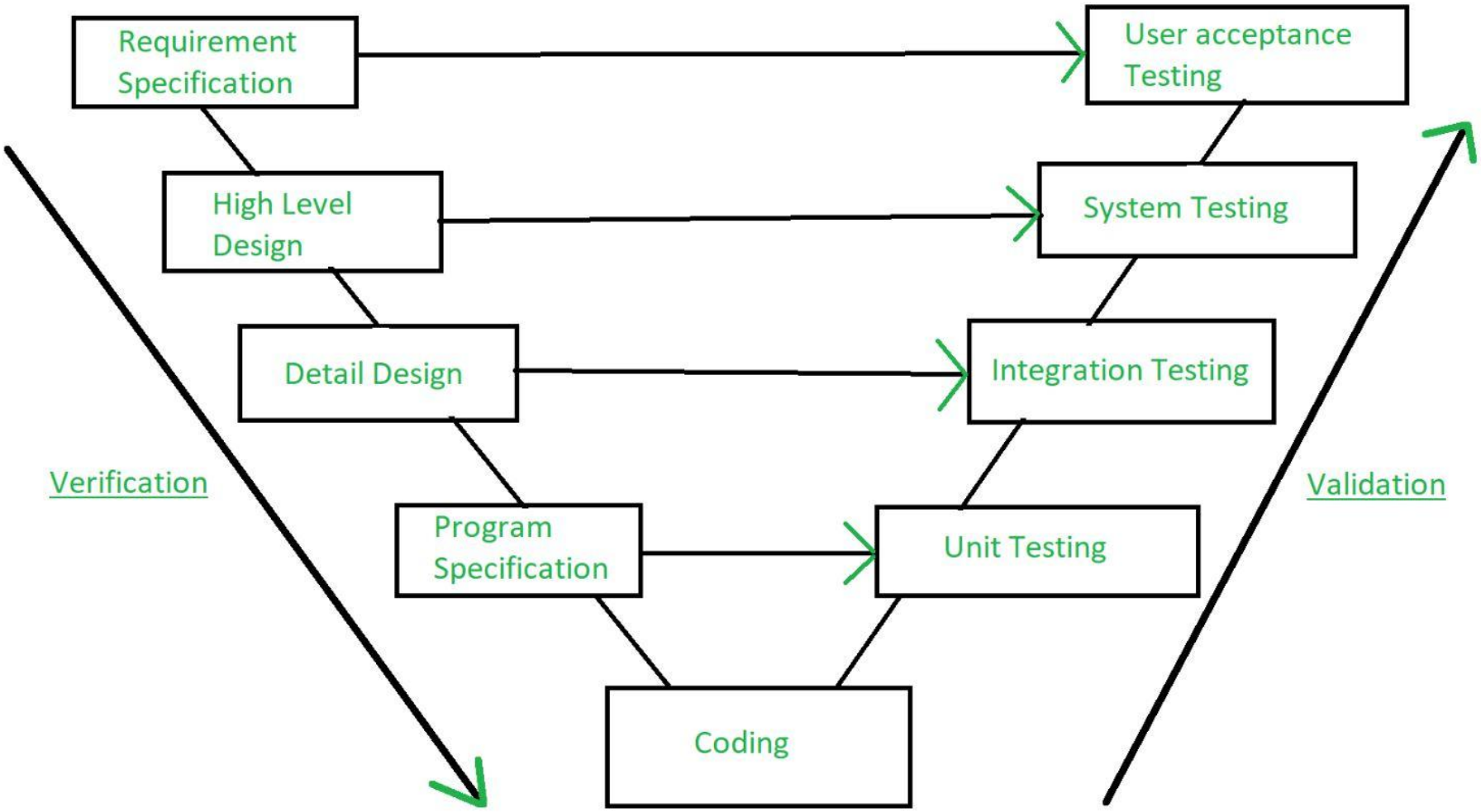
COMPUTER ENGINEERING

Deferences of Verification and Validation

Verification	Validation
Execution of code is not comes	Execution of code is comes
Explain whether the outputs are according to inputs or not.	Describe whether the product is accepted by the user or not.
Verification is carried out before the Validation.	**Validation** activity is carried out before the delivery.
Involved items: Plans, Requirement Specifications, Design Specifications, Code, Test Cases etc,	Involved items: Actual product under test, user specification.
Cost of errors caught in **Verification** is less than errors found in Validation.	Cost of errors caught in **Validation** is more than errors found in Verification.

COMPUTER ENGINEERING

Memory Hierarchies



COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Q&A