UESTC4019: Real-Time Computer Systems and Architecture

Lecture 2

Real-Time Systems

Part - 1

What is meant by Real-Time?

- Most people would probably understand “in real time” to mean “at once” or “instantaneously”.

- According to *the the Random House Dictionary of the English Language* (2nd unabridged edition, 1987),

  “Realtime” as pertaining to applications in which the computer must respond as rapidly as required by the user or necessitated by the process being controlled.

Examples

- Are these real-time systems?
  - An aircraft uses a sequence of accelerometer pulses to determine its position
  - Handling an overtemperature failure in a nuclear power plant
  - Consider a situation in which a passenger approaches an airline check-in counter to pick up his boarding pass for a certain flight from New York to Boston, which is leaving in five minutes

Software

- The hardware of a computer solves problems by repeated execution of machine - language instructions, collectively known as software

- Software, on the other hand, is traditionally divided into
  - system programs, and
  - application programs

System Programs

- System programs consist of software that interfaces with the underlying computer hardware, such as drivers, interrupt handlers, task schedulers, and various programs that act as tools for the development or analysis of application programs. Examples:
    - compilers, which translate high-level language programs into assembly code
    - assemblers, which convert the assembly code into a special binary format called object or machine code
    - linkers/locators, which prepare the object code for execution in a specific hardware environment

Operating System

- An operating system is a specialized collection of system programs that manage the physical resources of the computer

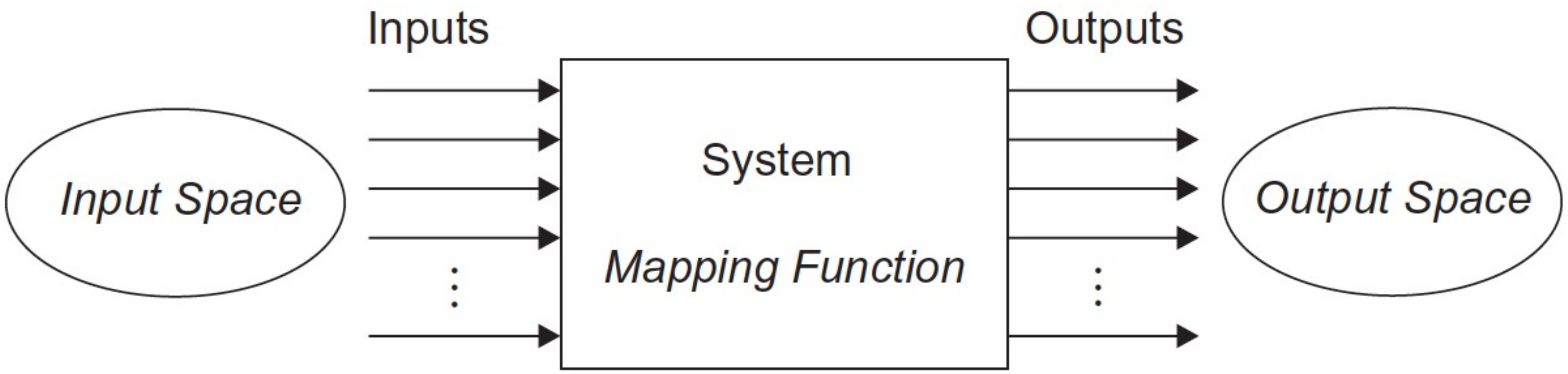
- As such, a real-time operating system (RTOS) is a truly important system program

Application Programs

- Application programs are programs written to solve specific problems, such as
  - optimal hall - call allocation of an elevator bank in a high-rise building,
  - inertial navigation of an aircraft, and
  - payroll preparation for some industrial company
- Certain design considerations play a role in the design of system programs and application software intended to run in real-time environments

System

- A system is a mapping of a set of inputs into a set of outputs
- The mapping function between input and output spaces can be considered as a black box with one or more inputs entering and one or more outputs exiting the system

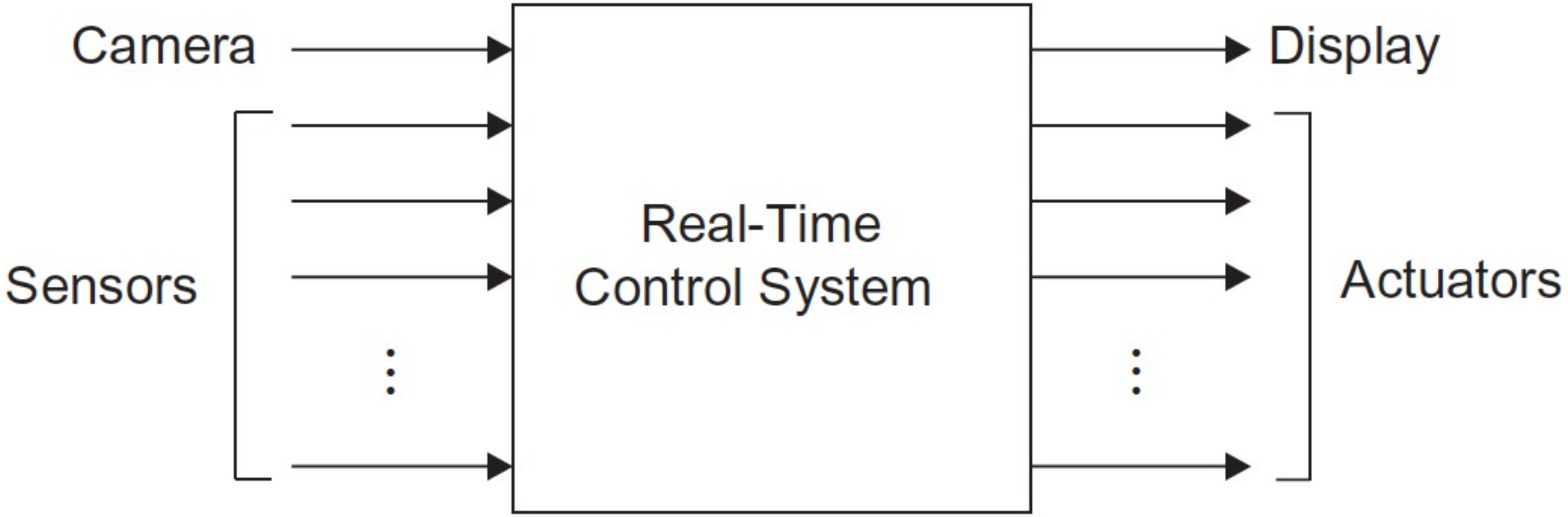


Properties of a System

- Vernon (Vernon, 1989) lists five general properties that belong to any “ system ”
  1. A system is an assembly of components connected together in an organized way
  2. A system is fundamentally altered if a component joins or leaves it
  3. It has a purpose
  4. It has a degree of permanence
  5. It has been defined as being of particular interest

P. Vernon , “ Systems in engineering , ” IEE Review , 35 (10), pp. 383 – 385 , 1989

System – An Example

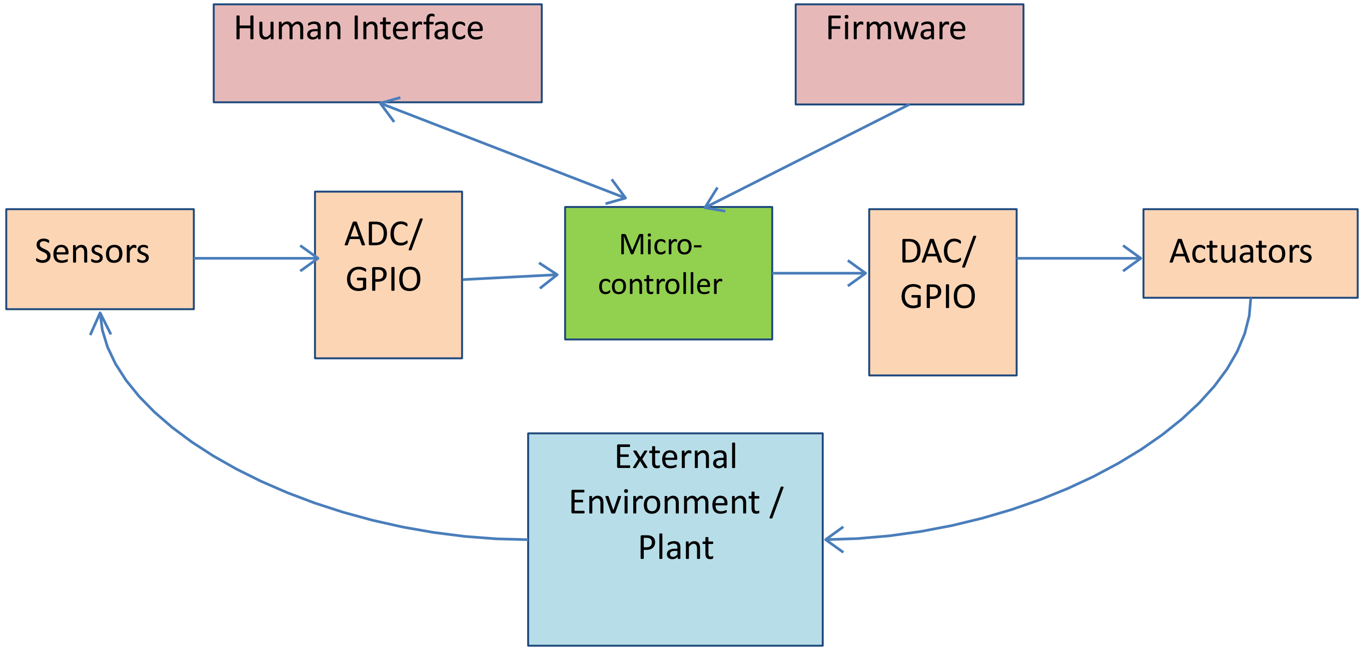


A real-time control system including inputs from a camera and multiple sensors, as well as outputs to a display and multiple actuators

System Input & Output

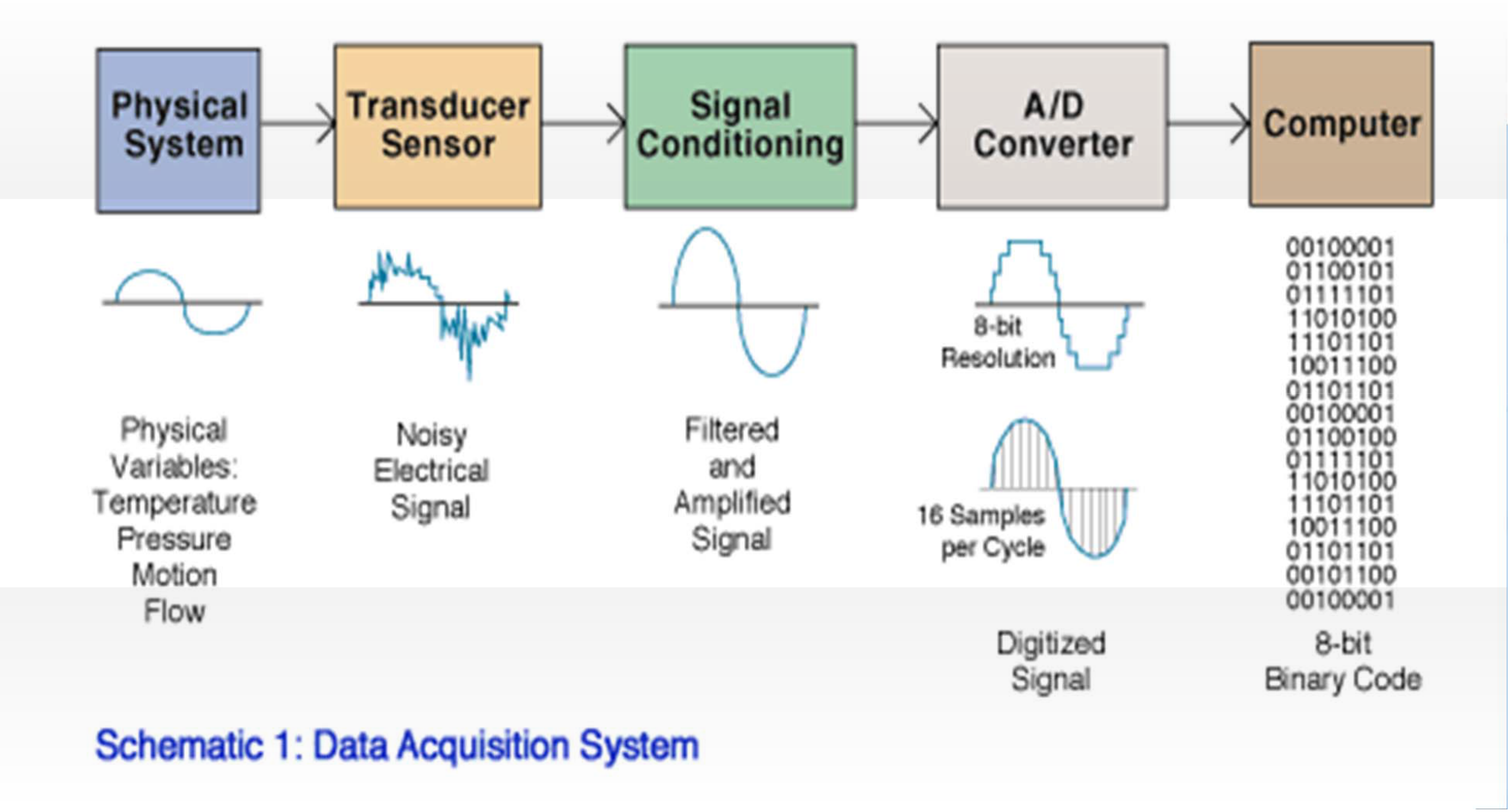
- In computing systems, the inputs represent digital data from hardware devices or other software systems.
- The inputs are often associated with sensors, cameras, and other devices that provide analog inputs, which are converted to digital data, or provide direct digital inputs.
- The digital outputs of computer systems, on the other hand, can be converted to analog outputs to control external hardware devices, such as actuators and displays, or used directly without any conversion.

A Real-Time Control System



ADC – Analogue to digital converter
GPIO – General purpose Input / Output –
Digital I/O, I2C, SPIO, CAN etc..
DAC – Digital to Analogue Converter
Firmware – Monitoring and control software store in non-volatile memory – such as ROM, flash etc ..
Human Interface – Display - LED/LCD
Input - touch screen/keypad.
Micro-controller – micro-processor + memory + I/O

Signal Processing in a Data Acquisition System



Part - 2

Response Time

- There is some inherent delay between presentation of the inputs (excitation) and appearance of the outputs (response).

- Response Time

  The time between the presentation of a set of inputs to a system and the realization of the required behavior, including the availability of all associated outputs, is called the response time of the system.

- How fast and punctual the response time needs to be depends on the characteristics and purpose of the specific system.

Real-Time Computing

- In computer science, real-time computing (RTC), or reactive computing, is the study of hardware and software systems that are subject to a "real-time constraint"—i.e., operational deadlines from event to system response.
- By contrast, a non real-time system is one for which there is no deadline, even if fast response or high performance is desired or preferred.
- *"The needs of real-time software are often addressed in the context of real-time operating systems, and synchronous programming languages, which provide frameworks on which to build real-time application software."* [Wikipedia]

Real-Time System

- A real-time system is a computer/embedded system that must satisfy bounded response-time constraints or risk severe consequences, including failure

- The system must satisfy deadline constraints in order to be correct

- A real-time system is one whose logical correctness is based on both the correctness of the outputs and their timeliness

Failed System

- Inability of the system to perform according to system specification

- A failed system is a system that cannot satisfy one or more of the requirements stipulated in the system requirements specification

- Because of this definition of failure, rigorous specification of the system operating criteria, including timing constraints, is necessary

Embedded System – Definition (1)

- An embedded system is a computer system designed to perform one or a few dedicated functions, often with real-time computing constraints.
- An embedded system is a system containing one or more computers (or processors) having a central role in the functionality of the system, but the system is not explicitly called a computer.
- An embedded system is a computing device with tightly coupled hardware and software integration that are designed to perform a dedicated function.

Embedded System – Definition (2)

- An embedded system is an information processing systems embedded into a larger product
- Usually encapsulated by the device it controls. Often hidden from view, without the awareness of their users. Most users doesn’t know that it includes processors
- An embedded system can be a sub-system within a larger system
- Multiple embedded systems can coexist in an embedding system

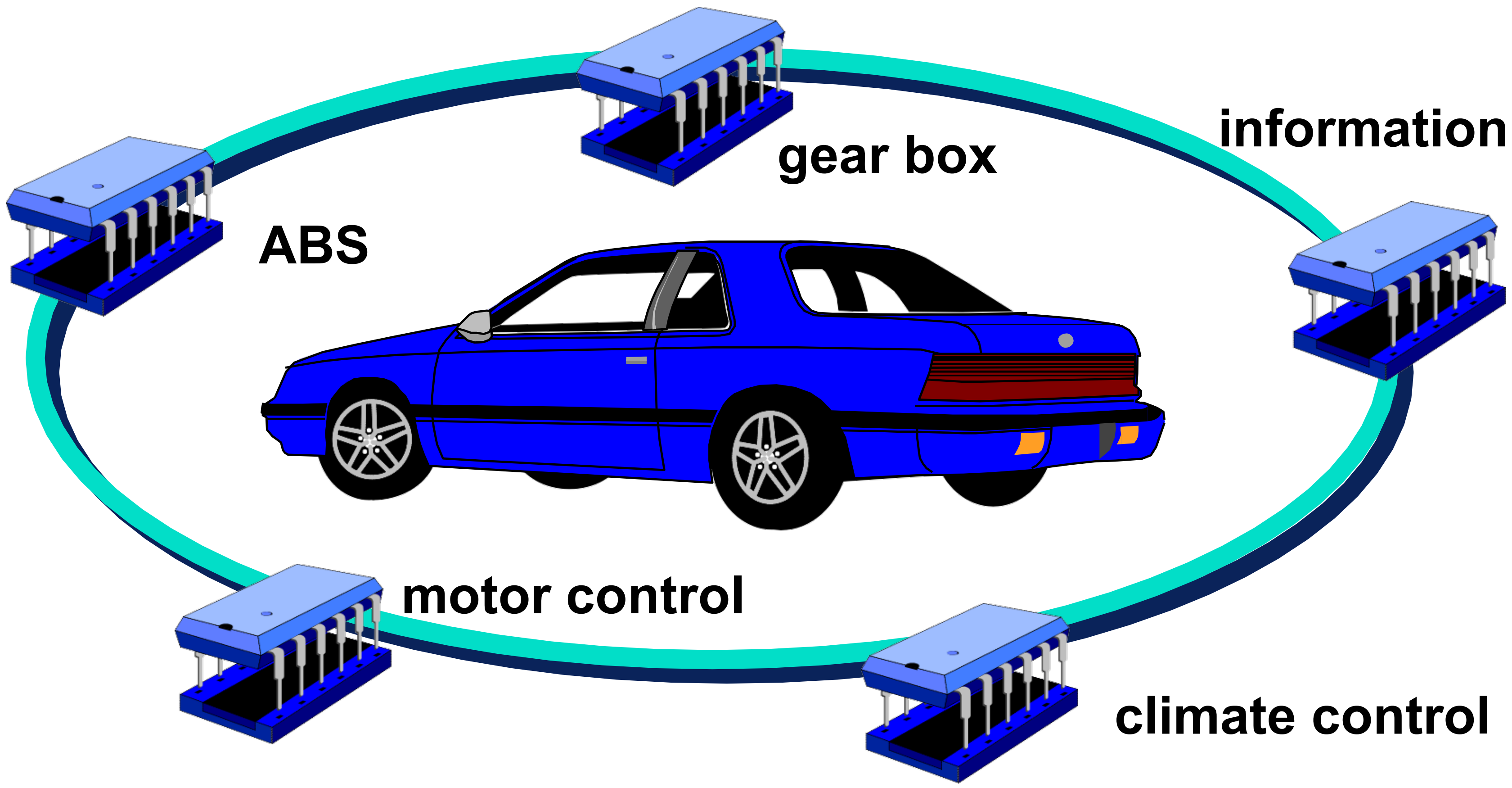
Embedded Computing

- The program instructions written for embedded systems are referred to as firmware, and are stored in read-only memory or Flash memory chips
- These instructions normally run with limited computer hardware resources: little memory, small or non-existent keyboard and/or screen
- Embedded processors can be microprocessors or microcontrollers.
- It is estimated that 99 % of all processors produced goes into embedded systems

Reactive System

- Reactive systems are those in which task scheduling is driven by ongoing interaction with their environment
- For example, a fire-control system reacts to certain buttons pressed by a pilot

An Example of Embedded System



Embedded Products

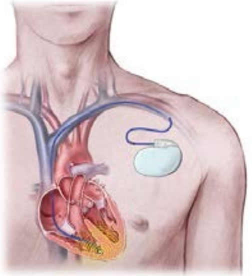
- Embedded products are found in
  - Industry
  - Automotive
  - Aerospace
  - Medical systems
  - Mobile systems
  - Communication
  - Networking
  - Household products (dishwasher, etc)
  - Media products – broadcasting
  - Cameras
  - ---- in other words, everythere ----

More Examples of Embedded Systems



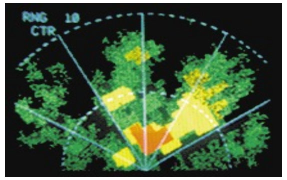






NASA Pathfinder (mission to MAR 1997)





Ground-proximity warning system

Part - 3

Back to the Examples

- Are these real-time systems?
  - An aircraft uses a sequence of accelerometer pulses to determine its position
  - Handling an overtemperature failure in a nuclear power plant
  - Consider a situation in which a passenger approaches an airline check-in counter to pick up his boarding pass for a certain flight from New York to Boston, which is leaving in five minutes
- A system does not have to process data at once or instantaneously to be considered real-time; it must simply have response times that are constrained appropriately

Types of Real-Time System

- A soft real-time system is one in which performance is degraded but not destroyed by failure to meet response-time constraints
- A hard real-time system is one in which failure to meet even a single deadline may lead to complete or catastrophic system failure
- A firm real-time system is one in which a few missed deadlines will not lead to total failure, but missing more than a few may lead to complete or catastrophic system failure

Examples of Real-Time Tasks

- Hard Real-Time Tasks

- safety-critical systems
- Sensory data acquisition
- data filtering and prediction
- detection of critical conditions
- data fusion and image processing
- actuator servoing
- low-level control of critical system components
- action planning for systems that tightly interact with the environment

- Soft Real-Time Tasks

- Video playing
- audio/video encoding and decoding
- on-line image processing
- sensory data transmission in distributed systems
- The command interpreter of the user interface
- handling input data from the keyboard
- displaying messages on the screen
- representation of system state variables

Real-Time Punctuality

- In cost - effective and robust real-time systems, a pragmatic rule of thumb could be: process everything as slowly as possible and repeat tasks as seldom as possible
- Real-time punctuality: every response time has an average value, t_R , with upper and lower bounds of $t_R + \varepsilon_U$ and $t_R - \varepsilon_L$, respectively, and ε_U , $\varepsilon_L \rightarrow 0^+$
- In all practical systems, the values of ε_U and ε_L are nonzero, though they may be very small or even negligible
- The nonzero values are due to cumulative latency and propagation-delay components in real-time hardware and software

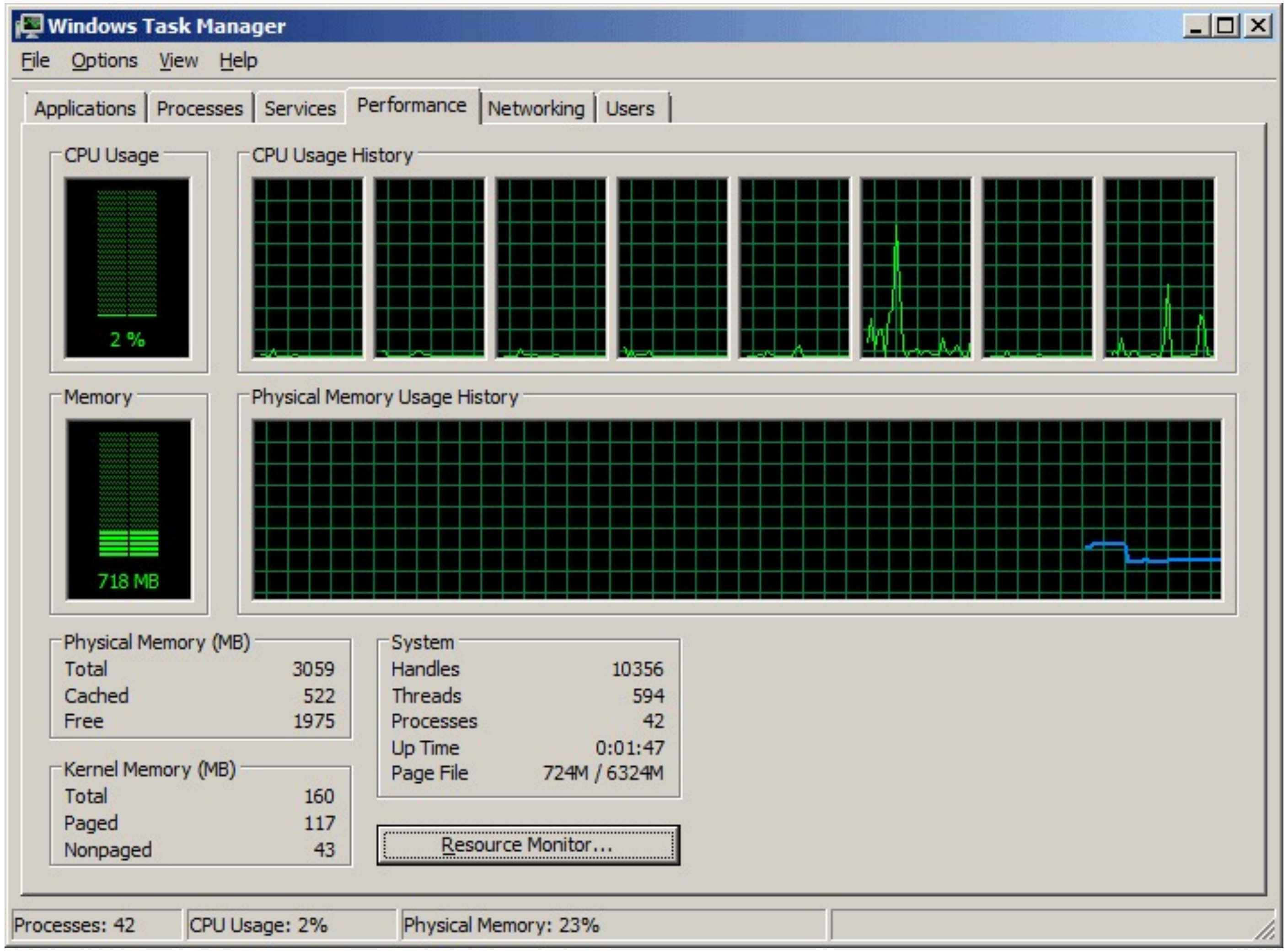
CPU Utilization Factor (1)

- The CPU utilization or time-loading factor, U , is a relative measure of the nonidle processing taking place
- A system is said to be time-overloaded if U > 100%
- Systems that are too highly utilized are problematic, because additions, changes, or corrections cannot be made to the system without risk of time-overloading
- Systems that are not sufficiently utilized are not necessarily cost-effective

CPU Utilization Factor (2)

- Utilization of
  - 50% is common for new products
  - 80% might be acceptable for systems that do not expect growth
  - 70% as a target for U is one of the most celebrated and potentially useful results in the theory of real-time systems where tasks are periodic and independent

CPU Utilization



Calculating Utilization Factor (1)

- The CPU utilization factor U is calculated by summing the contribution of utilization factors for each (periodic or aperiodic) task
- Suppose a system has $n \geq 1$ periodic tasks, each with an execution period of p_i , and hence, execution frequency, $f_i = 1/p_i$
- If task i is known to have (or has been estimated to have) a worst-case execution time of e_i , then the utilization factor, u_i , for task i is

$$u_i = e_i / p_i.$$

Calculating Utilization Factor (2)

- The overall system utilization factor can be calculated as:

$$U = \sum_{i=1}^{n} u_i = \sum_{i=1}^{n} e_i / p_i.$$

Event & Release Time

- Any occurrence that causes the program counter to change non sequentially is considered a change of flow-of-control, and thus an event.

- In scheduling theory, the release time of a job is similar to an event.

- The release time is the time at which an instance of a scheduled task is ready to run, and is generally associated with an interrupt.

- Events are slightly different from jobs in that events can be caused by interrupts, as well as branches.

Synchronous vs Asynchronous Events

- Synchronous events are those that occur at predictable times in the flow-of-control.
- The change in flow-of-control, represented by a conditional branch instruction, or by the occurrence of an internal trap interrupt, can be anticipated.
- Asynchronous events occur at unpredictable points in the flow-of-control and are usually caused by external sources.
- An engineer can never count on a clock ticking exactly at the rate specified, so any clock-driven event must be treated as asynchronous.

Deterministic System

- A system is deterministic, if for each possible state and each set of inputs, a unique set of outputs and next state of the system can be determined

- Event determinism means the next states and outputs of a system are known for each set of inputs that trigger events.

- Thus, a system that is deterministic is also event deterministic.