Introduction to Real Time Systems

Real-time system definition (1)

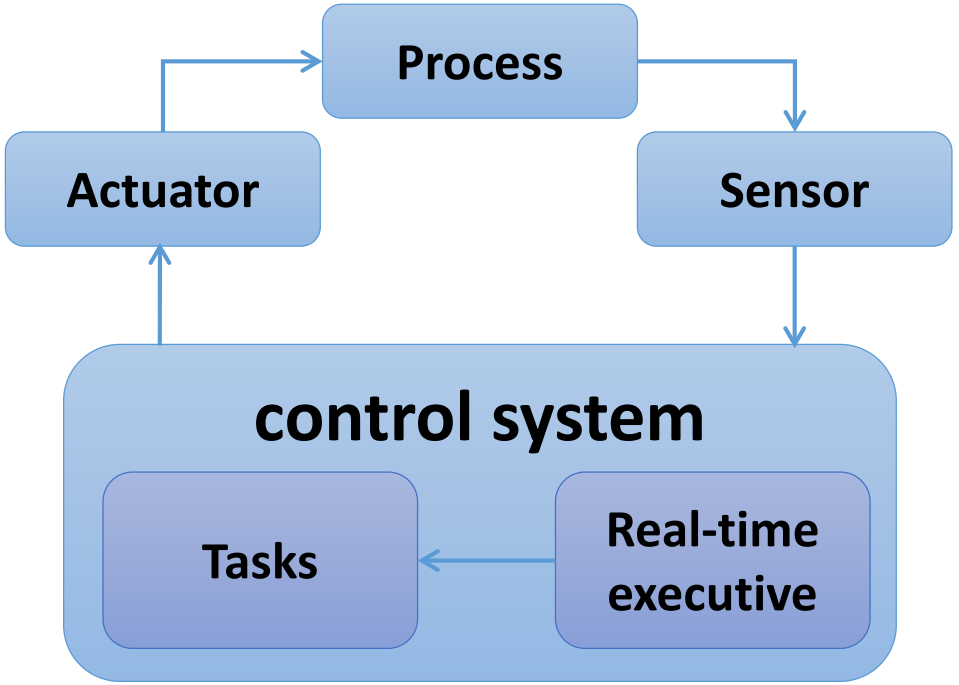
- **System:** set of “activities” corresponding to one or more tasks carried out in sequence or concurrently. The tasks possibly communicate with each other. The system interacts with its environment.
- **Real-time system (RTS):** system whose behavior depends not only on the accuracy of the processing carried out, but also on the time when the results of these processing are provided, i.e. that a delay in producing a result is considered an error.

Real-time system definition (2)

- A RTS is called critical if there are consequences of a system failure and concerns operational safety. Example: air traffic control.
- A RTS is subjected to time constraints, it is not necessarily fast. The time scale may vary depending on the context.
- Examples:
  - Chemical reaction (hours),
  - Manufacturing line (minutes),
  - Monitoring of nuclear power plants (seconds),
  - Tracking of a missile/Radar (microsecond)

Real-time system definition (3)

- In a real-time system, a calculation result that is mathematically exact but arrives beyond a predefined deadline is a false result.
- A real-time system is made up of a controlled system (process) and a control system:



Classification of real-time systems (1)

- Based on system/environment interaction:
  - **Synchronous systems:** At times determined by an internal time reference to the system: time-driven systems, programmed to execute actions/readings at regular intervals set by a timer (sequential operation).Example: Red lights.
  - **Asynchronous systems:** At times determined by the environment itself: it waits for requests and reacts to them: event driven system. Example: Cooling system of a power plant.

Classification of real-time systems (2)

- Based on timing constraints:
  - **Hard real-time system:** This type of system can never miss its deadline. Missing the deadline may have disastrous consequences. The usefulness of results produced by a hard real-time system decreases abruptly and may become negative if tardiness increases. Tardiness means how late a real-time system completes its task with respect to its deadline. Example: Flight controller system.
  - **Soft real-time system:** This type of system can miss its deadline occasionally with some acceptably low probability. Missing the deadline have no disastrous consequences. The usefulness of results produced by a soft real-time system decreases gradually with an increase in tardiness. Example: Telephone switches.
  - **Firm Real-Time Systems:** These are systems that lie between hard and soft real-time systems. In firm real-time systems, missing a deadline is tolerable, but the usefulness of the output decreases with time. Examples of firm real-time systems include online trading systems, online auction systems, and reservation systems.

Characteristics of a real-time system

- **Predictability:** A RTS must be designed such that its performance is defined in the worst case. The performance of the application must be defined in all possible cases so as to ensure compliance with time constraints. We're talking about a worst case. To do this, it is necessary to know precisely the task parameters: overall calculation time for each activity, frequency, wake-up date, etc.
- **Determinism:** Remove any uncertainty about the behavior of tasks:
  - In a Hard RTS: we seek to ensure that all deadlines are met.
  - In a soft RTS: minimize the average delay of tasks for example.
  - Sources of non-determinism:
    - Calculation load (variations in activity execution times),
    - Inputs/outputs (reaction time, duration of communications),
    - Interruptions (system reaction time),
    - Hardware or software faults and exceptions.
- **Reliability:** Minimize direct human intervention (on-board real-time systems), Fault-tolerant design, to guarantee the behavior of the system and its components.

Limitations of classic real time systems

- Scheduling policies aim at equitable sharing of execution time.
- Not suitable for more critical tasks than others,
- I/O management generates long waits (sometimes unbounded),
- the management of interruptions is not optimized,
- the virtual memory management mechanisms are not optimal,
- the timers which organize time do not have a fine enough resolution.

Structure of RTS

- An RTS is made up of two essential layers, the “hardware” layer and the “software” layer.
  - **Hardware:** can have several configurations depending on the importance of the application and the cost to invest. However, the basic structure is the same.
  - **Software:** must allow the execution of the tasks assigned to the computer, and respond in optimal time to all actions/readings.
- Hardware/software compromise:
  - Capacities and speed (Memory, CPU, etc...).
  - Algorithm (complexity).
  - Hardware management (scheduling, memory management, preemption).

Applications of Real-time System

- **Industrial application:** Real-time system has a vast and prominent role in modern industries. Systems are made real time based so that maximum and accurate output can be obtained. These system somehow lead to the better performance and high productivity in less time. Examples: Automated Car Assembly Plant, Chemical Plant etc.
- **Medical Science application:** Due to the introduction of real-time system in medical science, many lives are saved and treatment of complex diseases has been turned down to easier ways. Some of the examples of medical science applications are: Robot, MRI Scan, Radiation therapy etc.
- **Telecommunication applications:** Real-time systems have enabled the whole world to connect via a medium across internet. These systems make the people connect with each other in no time and feel the real environment of togetherness. Examples of telecommunication applications are: Video Conferencing, Cellular system etc.
- **Defense applications:** Defenses are able to produce missiles which have great destroying ability. All these systems are real-time system and it provides the system to attack and to defend. Some of the applications of defense using real time systems are: Missile guidance system, anti-missile system, Satellite missile system etc.

General information on operating systems

- **An operating system (OS):** is a set of specialized programs that allows the use of the hardware resources of one or more computers in an optimal and efficient manner. An OS provides a human-machine interface (HMI) allowing communication between the user and machines through different application software.

Roles of an OS:

- **Tasks management:** it manages the allocation of the processor between the different programs using a scheduling algorithm.
- **RAM management:** it manages the memory space allocated to each application. If there is insufficient physical memory, the operating system can create a memory area on the hard disk, called “virtual memory”.
- **Input/output management:** it allows you to unify and control program access to hardware resources through drivers (device managers).
- **Management of application execution.**
- **File management:** it manages reading and writing to the file system and access rights to files by users and applications.
- **Information management.**

Important concepts

- **Process:** Continuous sequence of operations, actions constituting the way of doing or processing something.
- **Program:** A set of instructions and data representing an algorithm that can be executed by a computer.
- **Instruction:** an elementary operation of a process in a computer architecture.
- **Macro-instruction:** complex instruction, defining operations composed from instructions in the base directory of a computer.
- **A multitasking (multithreaded):** when several tasks (processes) can be executed simultaneously.
- **A preemptive system:** when it has a scheduler (planner), which distributes, according to priority criteria, time between the different processes that request it.
- **Multiprocessing:** a technique which consists of operating several processes in parallel in order to obtain greater computing power or to increase the availability of the system (in the event of a processor failure). A multiprocessing system manages the sharing of memory between several processes, also it distributes the workload.

Real-time OS

- **Real-time OS:** is an OS operates reliably according to specific temporal constraints, that is to say it must be capable of delivering correct processing of information received at well-defined time intervals. Here are some examples of real-time operating systems:
  - RTLinux (hard Real Time Linux, USA);
  - QNX (the BlackBerry company, USA/Canada);
  - Micro-Itron (free, Japan).
  - RTEMS, FreeRTOS (free and lightweight).
- Note: Linux is not RT, it can be modified to be Real Time (PREEMPT-RT, Xenomai).

Differences between OS and RTOS

- OS (Operating system or operating system):
  - At startup, the OS takes control,
  - Compilation + Editing the links + Executing the program,
  - Multiprogramming, multi-user.
- RTOS (Real-time operating system):
  - Editing the application and RTOS links,
  - At startup, the application takes control and starts the RTOS,
  - RTOS and the application are strongly coupled,
  - No protection from the application → better performance,
  - Services limited to embedded systems → reduction in memory size,
  - RTOS configuration: file management, I/O drivers, memory management, Tools, etc.

Regular OS	Real-Time OS (RTOS)
Complex	**Simple**
Best effort	**Guaranteed response**
Fairness	**Strict Timing constraints**
Average Bandwidth (power)	**Minimum and maximum limits**
Unknown components	**Components are known**
Unpredictable behavior	**Predictable behavior**
Plug and play	**RTOS is upgradeable**

Function of an RTOS

- The main functions of a real-time operating system are:
  - Action on external devices (converters, sensor readers, valve controllers, etc.);
  - Taking into account real time, providing any response in a minimum time;
  - Reaction to external events, with the minimum of human intervention;
  - Reliable management of information allowing operation, even in the event of hardware failures.