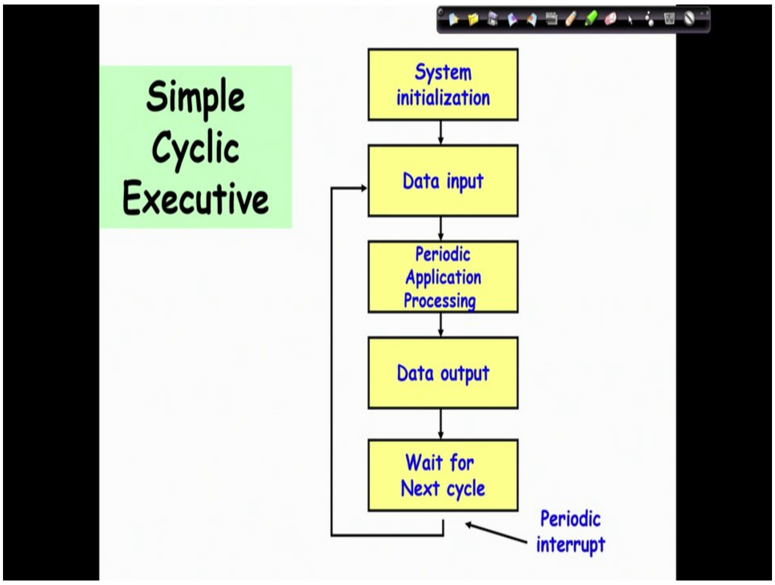
Real-time Operating System
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 03
Cyclic Executives

We had looked at some very basic issues in real-time operating systems and we had discussed that tasks scheduling is one of the major issues with the real-time operating system because the application deadlines are met with a help of a suitable tasks scheduler. Now, we will start looking at some of the schedulers that are being used. We will start with the simplest scheduler. It goes by the name of cyclic executives.

(Refer Slide Time: 00:59)



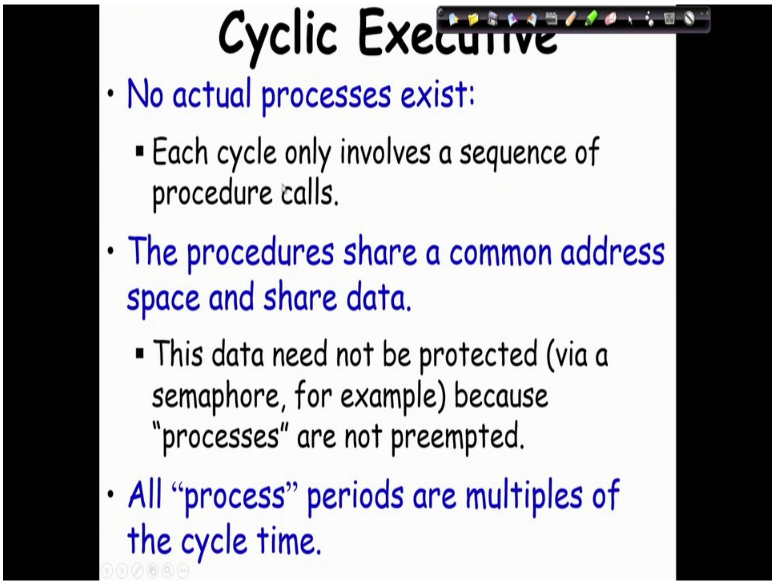
Simple Cyclic Executive

The cyclic executives are the simplest real-time operating systems. These are run on very simple embedded applications where there is a severe constraint and the processor capabilities may be a 4-bit or 8-bit processor and the memory is very less. In this situation a full-fledged operating system is difficult to use, the tasks are simple here and periodic in nature and this real-time cyclic executives are basically very small programs. Following is the basic structure of a cyclic executive.

Initially to start with the system, it is started with a system initialization where various parameters are set and then there is a periodic timer which gives timer interrupt and it is a cyclic executive which starts a call to a program or executes a program where initially there is a data input, application processing and the data output and then it keeps on waiting until the next period comes. This is the simplest real-time operating system. It's basically a small program. An example of its working is the data input.

Suppose we have a temperature controller. In the temperature controller initially, suppose the temperature has to be read from the sensor and this is the data input and then we need to check if it is below a threshold, higher than a threshold or is it abnormal etcetera and then based on that we possibly want to show the current temperature reading on a display. That is the data output and then we wait for the next interrupt and the cyclic executive continues looping here infinitely, each time waiting for the interrupt doing some simple processing. This is the simplest real-time operating system run on the most elementary embedded systems.

(Refer Slide Time: 03:50)

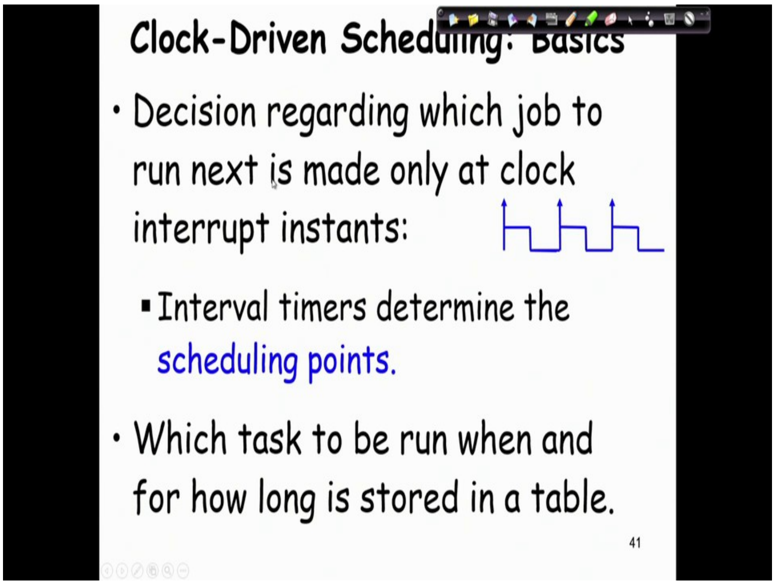


Cyclic Executive

- No actual processes exist:
  - Each cycle only involves a sequence of procedure calls.
- The procedures share a common address space and share data.
  - This data need not be protected (via a semaphore, for example) because "processes" are not preempted.
- All "process" periods are multiples of the cycle time.

In this cyclic executives, there are no processors. It basically involves running program. A program initially samples some sensors, does some processing and then calls some display and here this program is executed a number of times and each time there is an interrupt, this program is run.

(Refer Slide Time: 04:31)



Clock-Driven Scheduling: Basics

- Decision regarding which job to run next is made only at clock interrupt instants:
  - Interval timers determine the scheduling points.
- Which task to be run when and for how long is stored in a table.

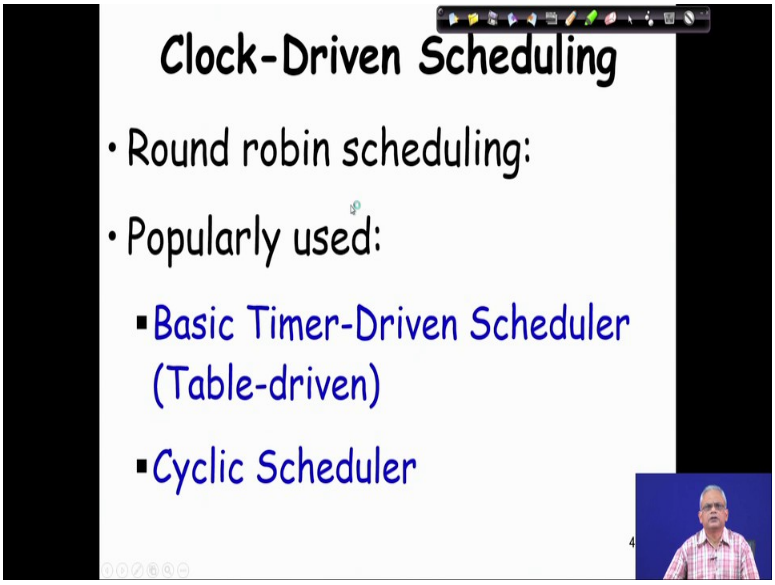
41

Looking at more examples of operating systems which are sophisticated compared to the simple cyclic executive. All these schedulers that we are discussing are at the simplest end of the real-time operating systems and these are called the clock driven schedulers. The ones that are much more sophisticated are the event-driven schedulers.

There are two broad categories of real-time task schedulers, one is the clock driven schedulers and the other one is the event-driven schedulers. At the simplest end of the real-time operating systems, we have the clock driven schedulers starting with simple cyclic executives. Following are slightly more sophisticated clock driven schedulers.

In all these clock driven schedulers the clock interrupt comes at regular intervals on the rising edge of the clock, the clock interrupts come and as soon as the clock interrupt comes, the scheduler becomes active. The points at which the clock interrupts come, they are called the scheduling points. And in our last discussion, we had mentioned that the instance at which the scheduler becomes active and checks what to do next is called as scheduling points and here the scheduling points are defined by a clock.

(Refer Slide Time: 06:27)



Clock-Driven Scheduling

- Round robin scheduling:
- Popularly used:
  - Basic Timer-Driven Scheduler (Table-driven)
  - Cyclic Scheduler