REAL TIME OPERATING SYSTEMS

Lesson-8:
I/O Subsystems
1. I/O Subsystem
I/O ports

- Subsystems of OS device management system—UART access sub system and the parallel port access sub system.
- They are used by drivers to communicate with the many devices that use them.
- I/O instructions depend on the hardware platform.
- I/O systems differ in different OSes.
I/O Subsystem in a Typical I/O System at in an OS

- Application
- IO Basic functions
- IO device driver functions
- Device Hardware or Port or IO Interface card
2. I/O Operations
OSes differing IO operations

- Traditional OSes *Synchronous IO operations*— at a certain fixed data transfer rates.
- RTOSes *Asynchronous IO operations* — at the variable data transfer rates.
Synchronous IO operations

- Are at a certain fixed data transfer rates.
- Therefore, a task (process) blocks till completion of the IO.
- For example, a write function, write( ) for 1 kB data transfer to a buffer.
Synchronous IO

- Synchronous IO operation means once synchronous IO initiates, the data transfer will block the task till 1 kB data gets transferred to the buffer.
- Similarly, read() once initiated blocks the task till 1 kB is read
Asynchronous IO operations

- Variable data transfer rates.
- Permits that a process of high priority to run and should not block during the IOs.
2. POSIX asynchronous functions for IOs
POSIX asynchronous functions for IOs

- aio_read()
- aio_write()
- aio_list()
- aio_error()
- aio_cancel,
- aio_suspend()—Suspension is till the next port-device interruption or till a timed out.
- aio_return() returns the status of completed operations.
Summary
We learnt

- I/O subsystems are part of OS services.
- Examples are UART access and parallel port access.
- Synchronous and asynchronous IOs.
- A task gets blocked during the synchronous IOs, for example, fread() or write().
- RTOSes support asynchronous IOs, for example, aio_read() and aio_write also in order to not to block a task during the IOs.
End of Lesson 8 of Chapter 8