Faculty Of Computers and Information
Mansoura University

Real-Time Operating Systems

Dr. Naira Elazab

Information Technology Dept.,
Faculty Of Computers and Information,
Mansoura University

ARCHITECTURAL ADVANCEMENTS

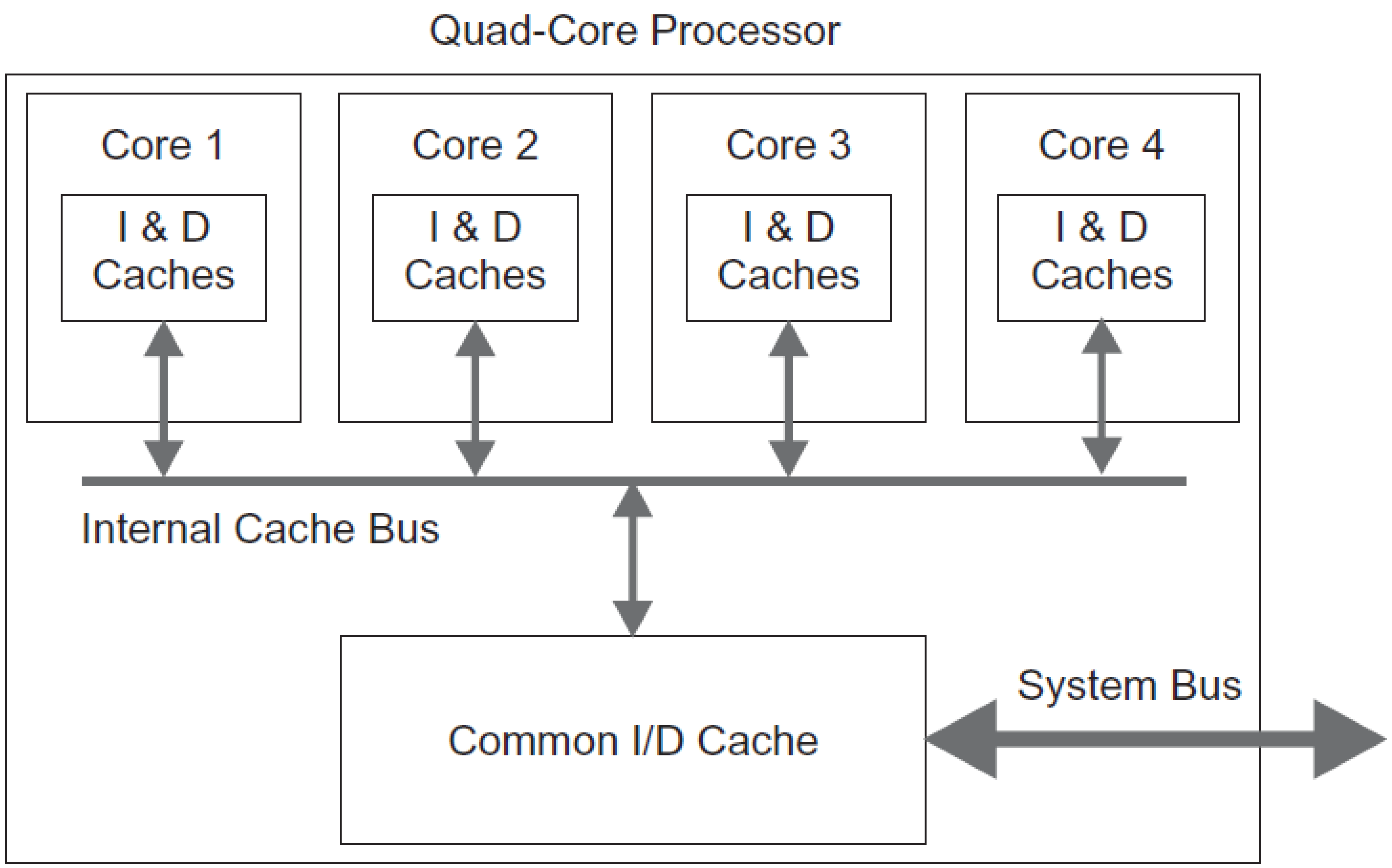
Modern Microprocessors

Increase clock speed to higher rates

Include multiple cores

- Pipelined Instruction Processing.
- Multi - Core Processors with Multi Tasks (i.e. the term " multi - core " refers to processors with two (dual - core), four (quad - core), or eight identical cores.
- I/O Interfacing (I/O polling, Interrupt – Driven I/O, DMA).

ARCHITECTURAL ADVANCEMENTS



MODERN MICROPROCESSORS (FLYNN'S TAXONOMY)

- Up to 80486, single stage pipeline
- Multiple stage pipeline, run many operations in parallel.
- Flynn's taxonomy is a classification of computer architectures, proposed by Michael J. Flynn in 1966.
- It has been used as a tool in design of modern processors and their functionalities.

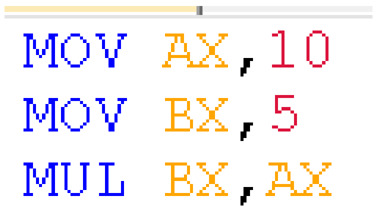
MODERN MICROPROCESSORS (FLYNN'S TAXONOMY)

- Single instruction stream single data stream (SISD)
- Single instruction stream, multiple data streams (SIMD), i.e. GPU
- Multiple instruction streams, single data stream (MISD), i.e. Space shuttle flight control computer
- Multiple instruction streams, multiple data streams (MIMD) i.e. multi-core processors and distributed systems.
- Single instruction, multiple threads (SIMT), SIMD + multithreading.
- Most of these architectures are extension to the x86 architecture.

MICROPROCESSORS ARCHITECTURE (RISC VS. CISC)

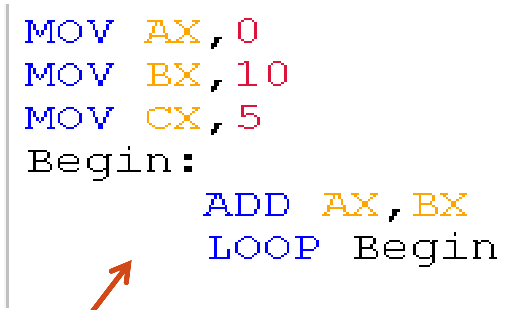
- Reduced Instruction Set Computing (RISC).
- Complex Instruction Set Computing (CISC).
- The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction.
- RISC does the opposite, reducing the cycles per instruction at the cost of increasing the number of instructions per program.

MICROPROCESSORS ARCHITECTURE (RISC VS. CISC)



```
MOV  AX,10
MOV  BX,5
MUL  BX,AX
```

CISC



```
MOV  AX,0
MOV  BX,10
MOV  CX,5
Begin:
        ADD  AX,BX
        LOOP Begin
```

RISC

The total clock cycles for the CISC version might be:

`(2 movs × 1 cycle) + (1 mul × 30 cycles) = 32 cycles`

While the clock cycles for the RISC version is:

`(3 movs × 1 cycle) + (5 adds × 1 cycle) + (5 loops × 1 cycle) = 13 cycles`

MICROPROCESSORS ARCHITECTURE (RISC VS. CISC)

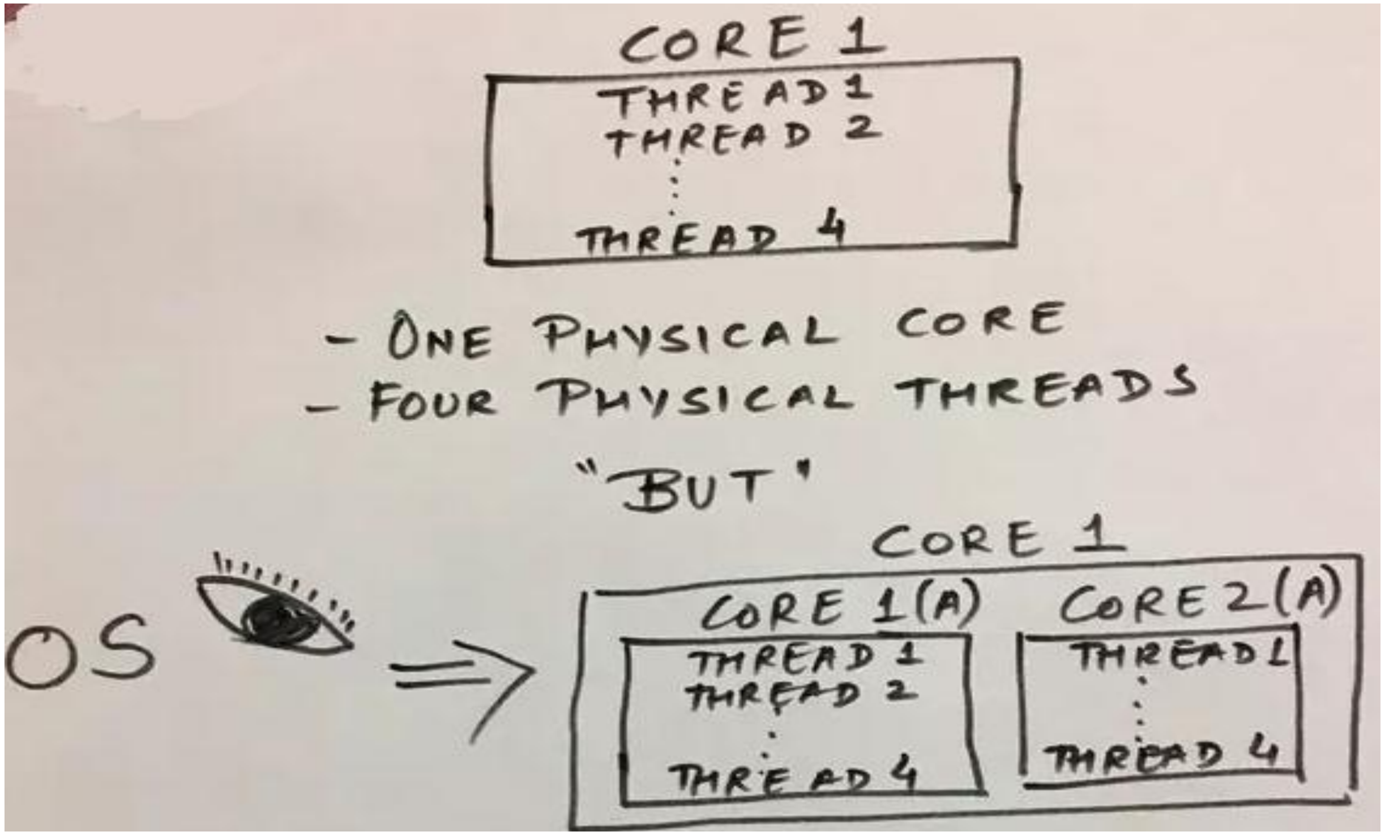
- The basic computer performance equation:

$$\text{CPU Time} = \frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{avg. cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

MICROPROCESSORS ARCHITECTURE (RISC VS. CISC)

RISC	CISC
Low cycles per second	High cycles per second
Instructions are simple (ADD, SUB, etc.)	Instructions are complex (MUL, DIV, etc.)
Required large memory	Required complex decoding circuits
Only load and store instructions can access memory.	Many instructions can access the memory.
Single clock instructions	Multi clock instructions.

MODERN MICROPROCESSORS (MULTITHREADING VS. HYPER-THREADING)



Operating System

- **Operating System controls resources:**
  - **Who gets the CPU;**
  - **When I/O takes place;**
  - **How much memory is allocated;**
- **Application programs run on top of OS services.**
- **Challenge: manage multiple, concurrent tasks.**

Operating System

- **OS Kernel**: is a computer program that is the core of a computer's operating system, with complete control over everything in the system.

- **Monolithic Kernel**: all the basic system services like process and memory management, interrupt handling etc were packaged into a single module in kernel space.

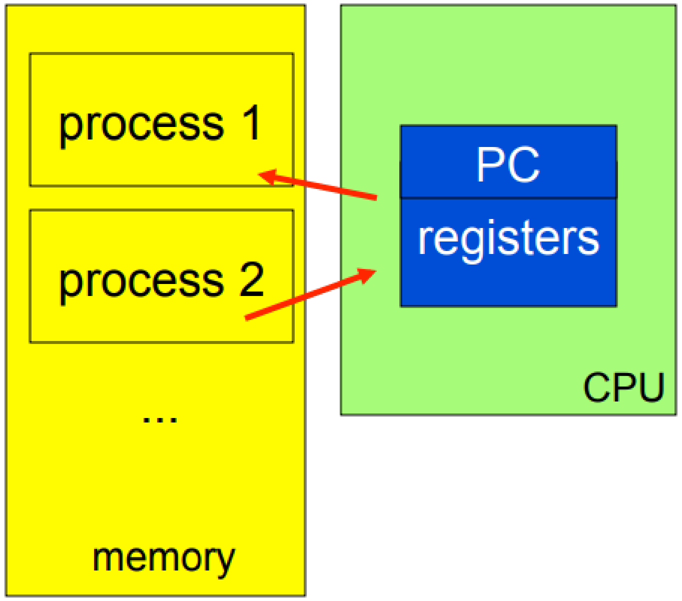
- **Microkernel**: the process resides in user-space in the form of servers. There is the server for managing memory issues, one server does process management, another one manages drivers, and so on.

Process

- A **process** is a unique execution of a program.
- A **process** has its own context:
  - ✓ Data in registers, PC, status.
  - ✓ Stored in Process Control Block (PCB)
- **Thread**: lightweight process
  - ✓ Threads share memory space in a same process.
- **OS** manages processes and threads.

Process

- A **Context Switch** is the process of storing the state of a process or of a thread, so that it can be restored and execution resumed from the same point later.
- Context Switch is used to support (Multitasking, Interrupt Handling, User and Kernel Mode Switching).



Kernel mode Vs. User mode

- In Kernel mode, the executing code has complete and unrestricted access to the underlying hardware. It can execute any CPU instruction and reference any memory address..

- In User mode, the executing code has no ability to directly access hardware or reference memory. Code running in user mode must delegate to system APIs to access hardware or memory.

Timing parameters of a (process | job | task | thread) Jj

- Arrival time (aj) or release time (rj) is the time at which the job becomes ready for execution.
- Computation (execution) time (Cj) is the time necessary to the processor for executing the job without interruption.
- Absolute deadline (dj) is the time at which the job should be completed.
- Relative deadline (Dj) is the time length between the arrival time and the absolute deadline.

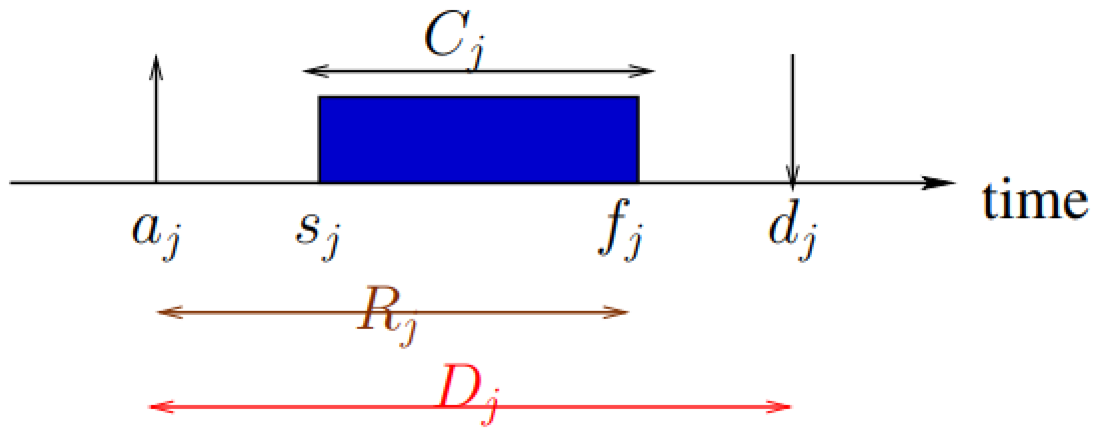
Timing parameters of a job J_j

- **Start time (sj) is the time at which the job starts its execution.**

- **Finishing time (fj) is the time at which the job finishes its execution.**

- **Response time (Rj) is the time length at which the job finishes its execution after its arrival, which is fj − aj .**

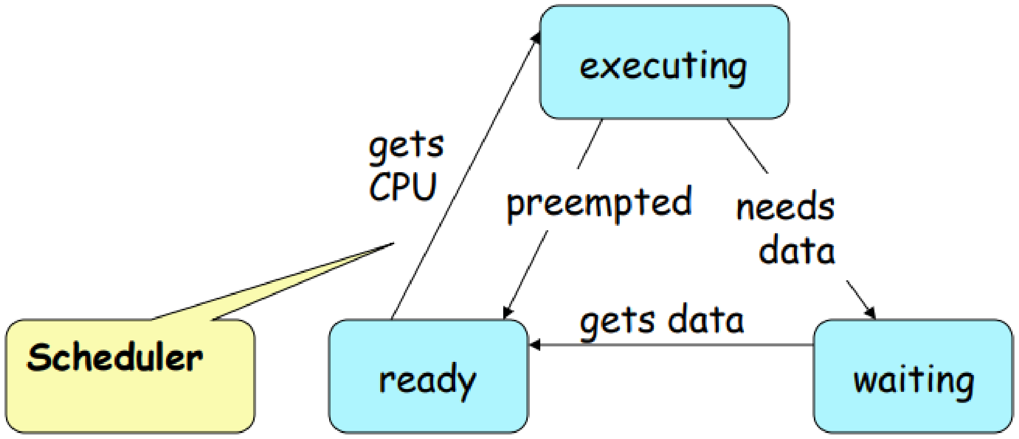
Timing parameters of a job J_j



Scheduling Concepts

- Scheduling Algorithm: determines the order that jobs execute on the processor.

- A process can be in one of three states:
  - ❑Executing on the CPU;
  - ❑Ready to run;
  - ❑Waiting for data.

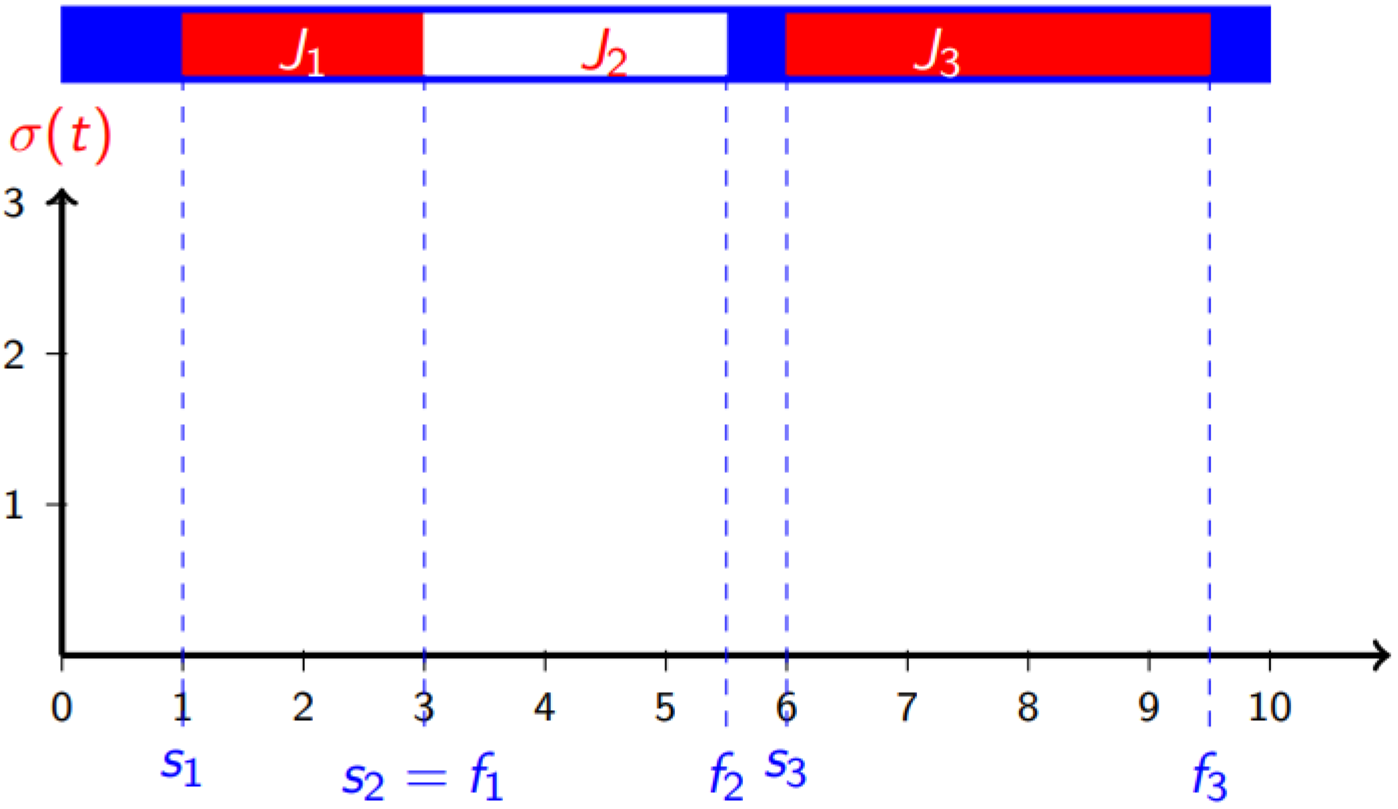


Schedules for a set of jobs {J1, J2, . . . , JN}

- A schedule is an assignment of jobs to the processor, such that each job is executed until completion.
- A schedule can be defined as an integer step function σ : R → N, where σ(t) = j denotes job Jj is executed at time t.
- Non-preemptive scheduling: there is only one interval with σ(t) = j for every Jj , where t is covered by the interval.
- Preemptive scheduling: there could be more than one interval with σ(t) = j.

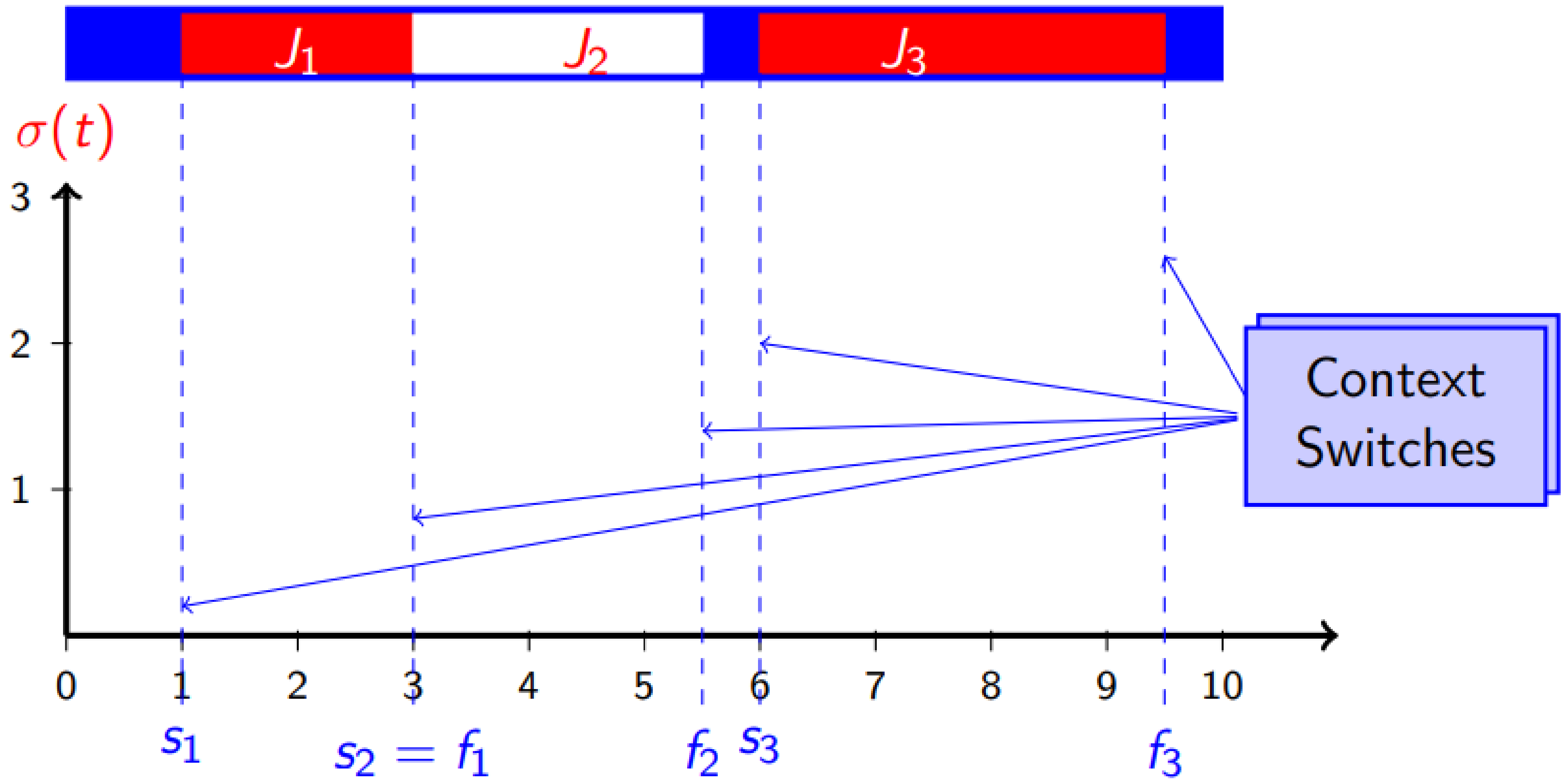
Scheduling Concept: Non-preemptive

Schedule: $\sigma : \mathbb{R} \to \mathbb{N}$ function of processor time to jobs



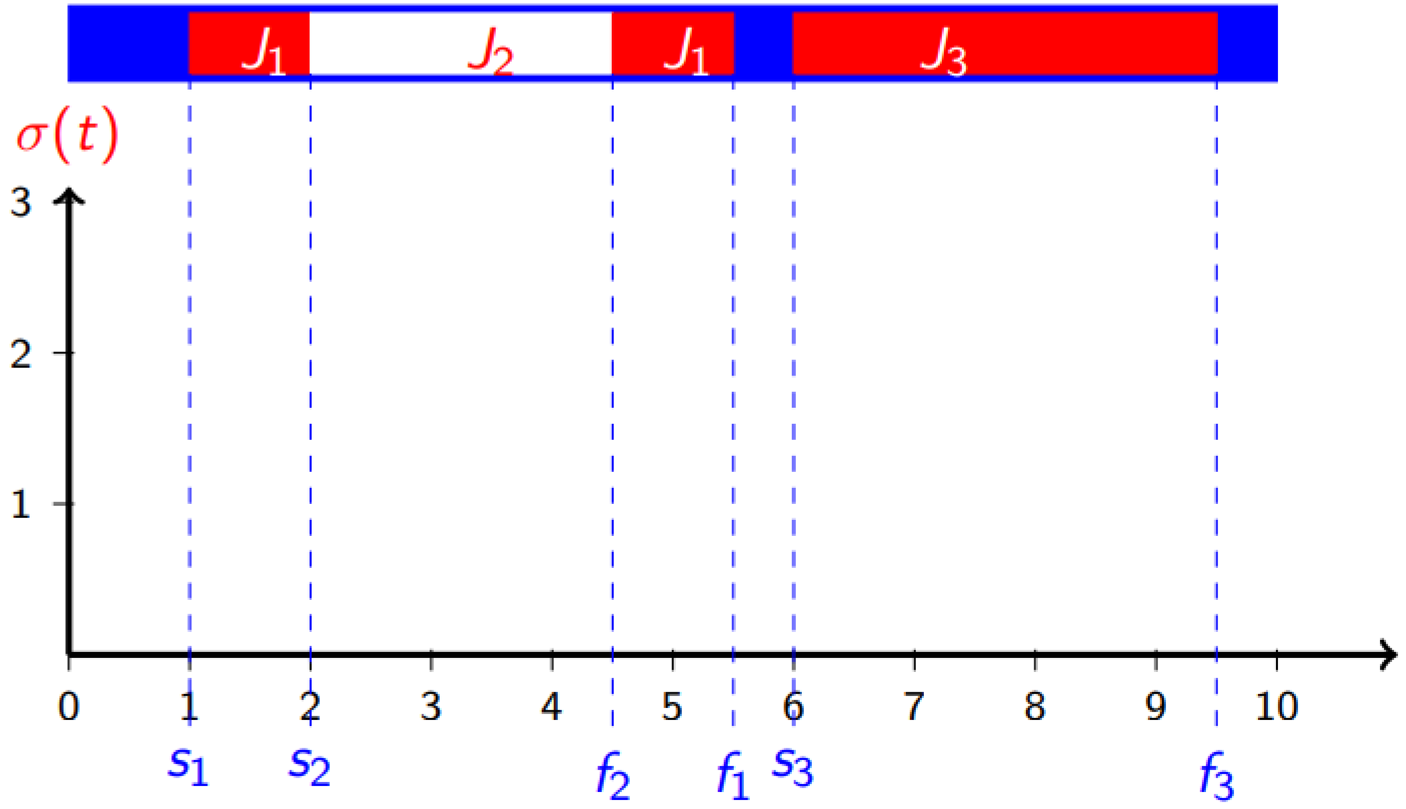
Scheduling Concept: Non-preemptive

Schedule: $\sigma : \mathbb{R} \rightarrow \mathbb{N}$ function of processor time to jobs



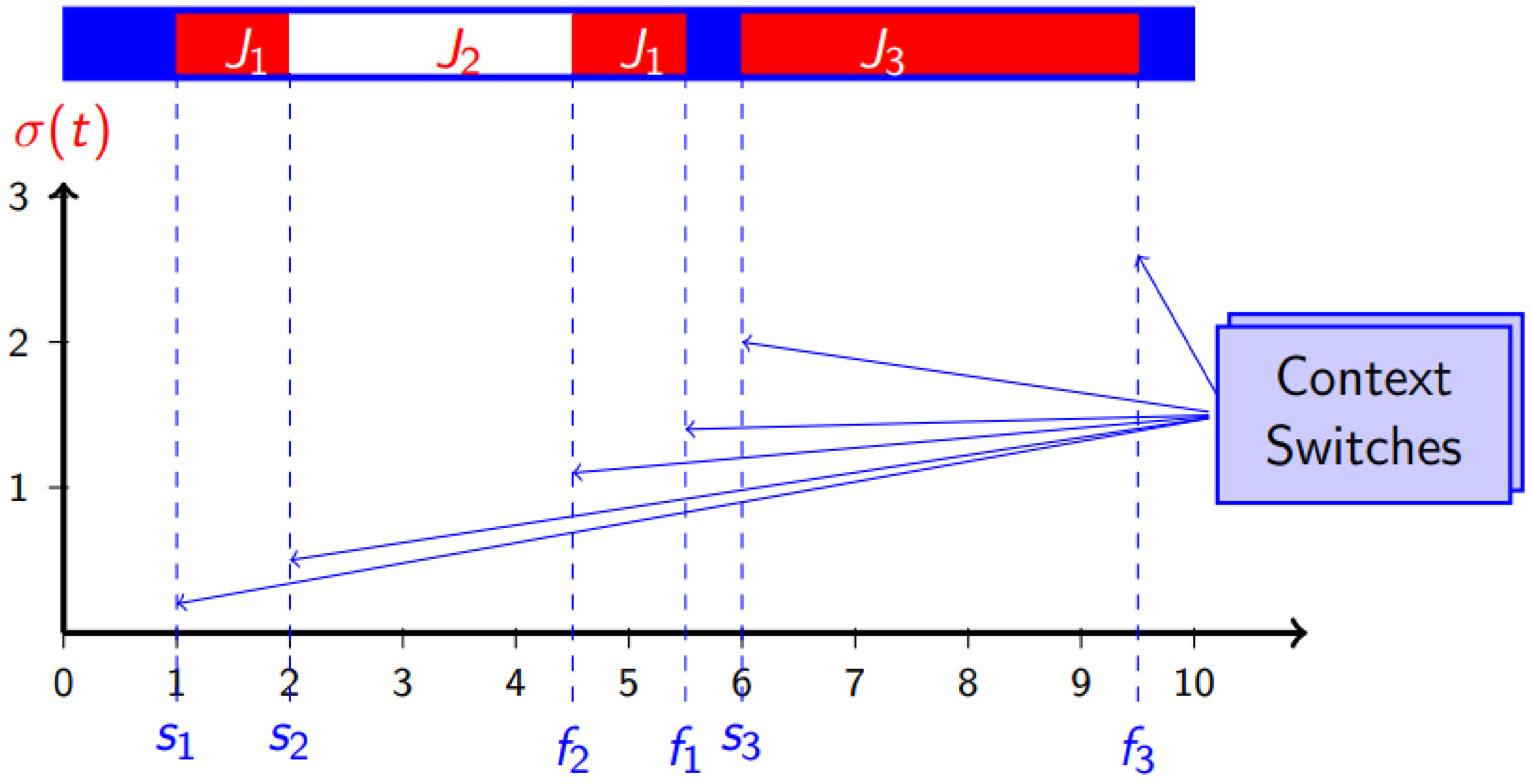
Scheduling Concept: Preemptive

Schedule: $\sigma : \mathbb{R} \rightarrow \mathbb{N}$ function of processor time to jobs



Scheduling Concept: Preemptive

Schedule: $\sigma : \mathbb{R} \rightarrow \mathbb{N}$ function of processor time to jobs



J_1
J_2
J_1
J_3
$\sigma(t)$
3
2
1
0
1
2
3
4
5
6
7
8
9
10
s_1
s_2
f_2
f_1
s_3
f_3
Context Switches