COE718: Embedded Systems Design

Lecture 8:
Real-Time Scheduling

Anita Tino

Scheduling Up Until Now…

- Revolves around arrival times and service times
- However what else do we require in real-time systems?

Scheduling Up Until Now…

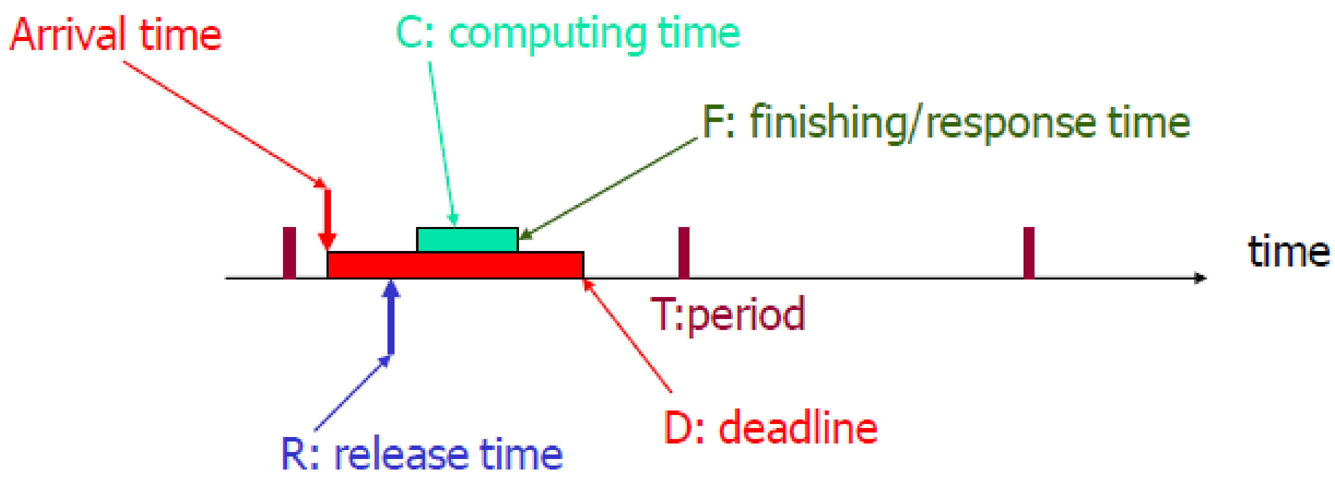
- Revolves around arrival times and service times
- However what else do we require in real-time systems?
- Need to take into consideration:
  - Deadlines!
  - CPU utilization

Real-Time Scheduling

- 2 types of process/task initiations:

1. **Periodic Tasks** – execute on (almost) every period
   - Don't confuse this with Round Robin
2. **Aperiodic Tasks**– execute on demand
   - Hard, must consider worst-case combinations in which the tasks may activate

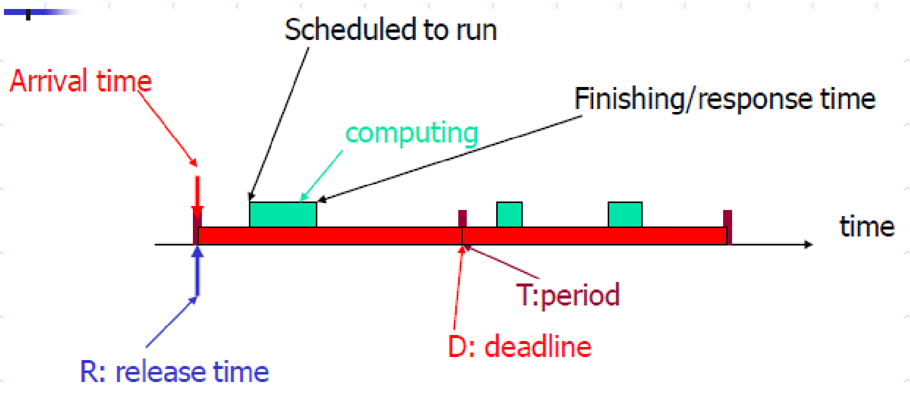
Scheduling Periodic Tasks



Scheduling Periodic Tasks

Each task:

- is released at a constant rate, with period T
- has worst case execution time C
  - Therefore C <= T
- has an arrival time A
- has release time R
  - ideally task would like to be released on arrival, i.e. R= A
- has a relative deadline time D



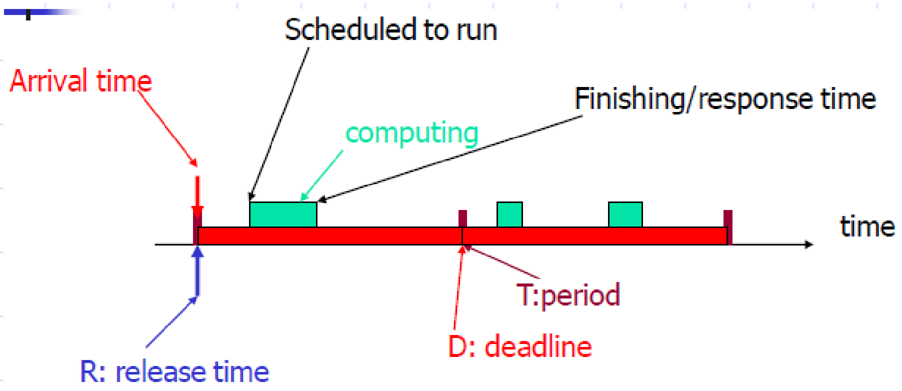
• A task may be represented as (C, T), and a task set as (Ci, Ti), which assumes all tasks are independent

Scheduling Periodic Tasks

For real-time scheduling...

- Deadline
  - D=T (ideally)
- CPU Utilization
  - U = C/T
  - For a task set:
    - $U = \sum C_i/T_i$
  - If U > 1, CPU is overloaded, tasks will fail to meet their deadlines no matter what scheduling technique is used
  - If U <=1, scheduling of tasks on CPU is possible, but will depend on scheduling algorithm used

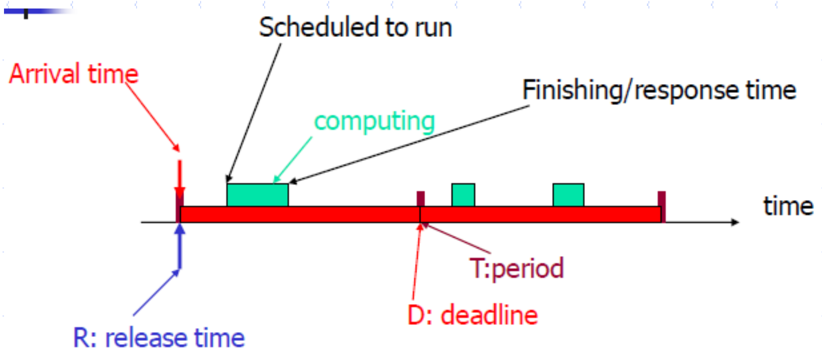
- A task may be represented as (C, T), and a task set as (Ci, Ti), which assumes all tasks are independent



Real-Time Scheduling Algorithms

- Fixed Priority Scheduling (FPS)
  - Rate Monotonic Scheduling (RMS)
  - Earliest Deadline First (EDF)
  - .....

- Priority Inversion



Fixed Priority Scheduling (FPS)

- Most widely used approach
- As we have been doing so far, each task has a static and fixed priority that is computed before execution
- The processes are executed in order of priority
- “Priority” in RT scheduling = temporal requirements
  - Priority in RT != correct functioning or integrity of system (as we’ve been doing for generic scheduling)

Fixed Priority Scheduling (FPS)

- Each method will have a schedulability test

U = formula <= #

-Pre-calculated number
-unique per algorithm

Σ Ci/Ti

Sum all the task utilizations

If the summations is not less than or equal to this #, the scheduling method will likely fail in trying to schedule the tasks to the processor

Fixed Priority Scheduling (FPS)

- Each method will have a schedulability test

U = formula <= #

-Pre-calculated number
-unique per algorithm

$\sum$ Ci/Ti

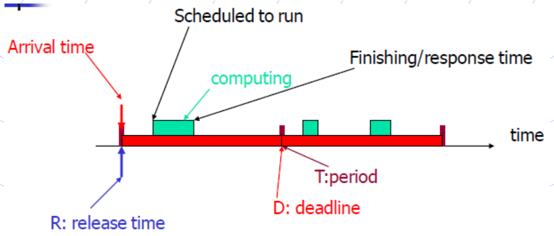
Sum all the task utilizations

If the summations is not less than or equal to this #, the scheduling method will likely fail in trying to schedule the tasks to the processor

-Schedulability test is said to be sufficient, but not necessary
-Are certain cases where it will actually schedule if > (assuming U <= 1)

Rate Monotonic Scheduling (RMS)

- Assumes periodic tasks, where Di = Ti
- Tasks are always released at the start of their periods
- Tasks are independent
- Each task has a unique priority based on it’s period
- Tasks with smaller periods get higher priority
- Priority #1 is the lowest



Rate Monotonic Scheduling (RMS)

- Schedulability test

$$U = \sum_{i=1}^{N} \frac{C_i}{T_i} \leq N(2^{1/N} - 1)$$

- C = computation time
  - Worst Case Execution Time (WCET)
- T = min time b/w process releases
- N = number of processes

Some calculations and examples for RMS…

Earliest Deadline First (EDF)

- Consists of a set of periodic tasks
- Do not all arrive at t= 0
  - Have different arrival times
- Whenever a ready task arrives, the ready queue is sorted so that the task closest to the end of its deadline is assigned the highest priority
  - May preempt the running task if necessary

Earliest Deadline First (EDF)

- Schedulability test

$$\sum_{i=1}^{N} \frac{C_i}{T_i} \leq 1$$

- C = computation time
  - Worst Case Execution Time (WCET)
- T = min time b/w process releases
- Must be less than or equal to 1
  - What does this mean?

Earliest Deadline First (EDF)

Advantages

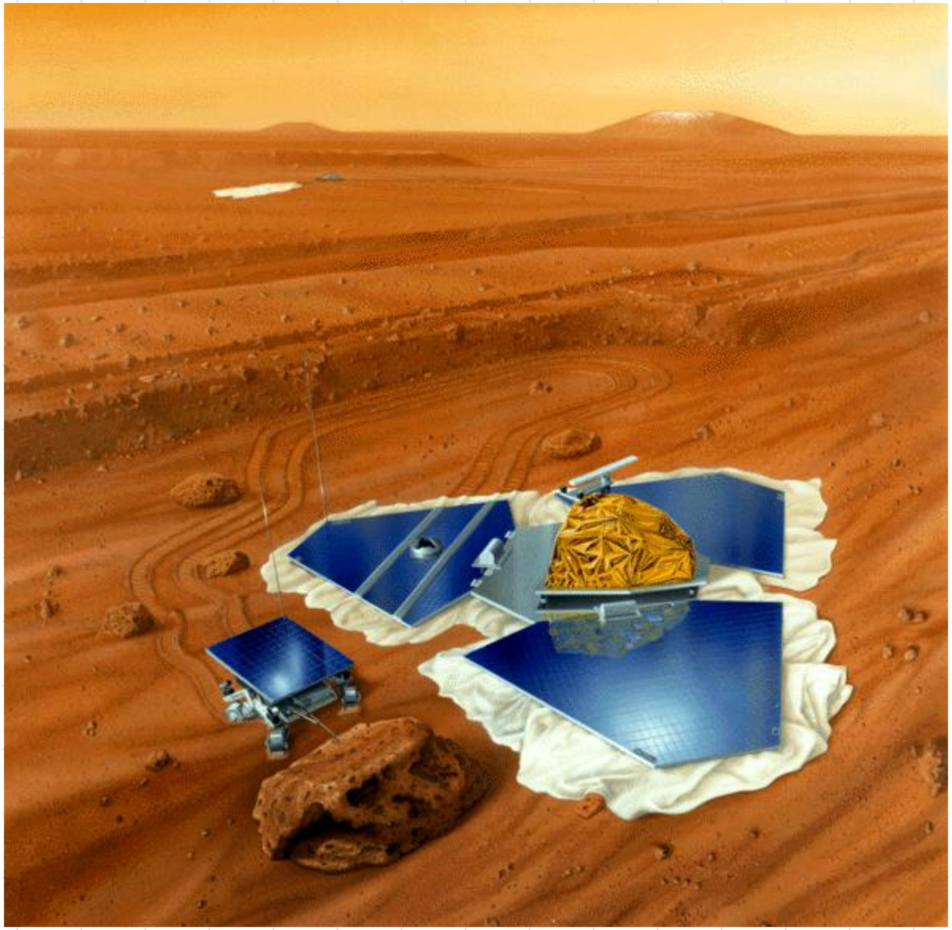
- Simple and works nicely in theory
- Easy schedulability test
- Optimal
- Superior to FPS in terms of supporting high utilizations

Disadvantages

- Requires a more complex runtime system with higher overhead
- Dynamic nature leads to unpredictablility
  - A domino effect can occur when a large number of tasks end up missing their deadlines

Some examples for EDF...

Mars Pathfinder Problem



3 tasks:

1. Low priority
2. Medium priority
3. High priority

-Low and high tasks share a bus
-Must obtain a mutex to access the bus

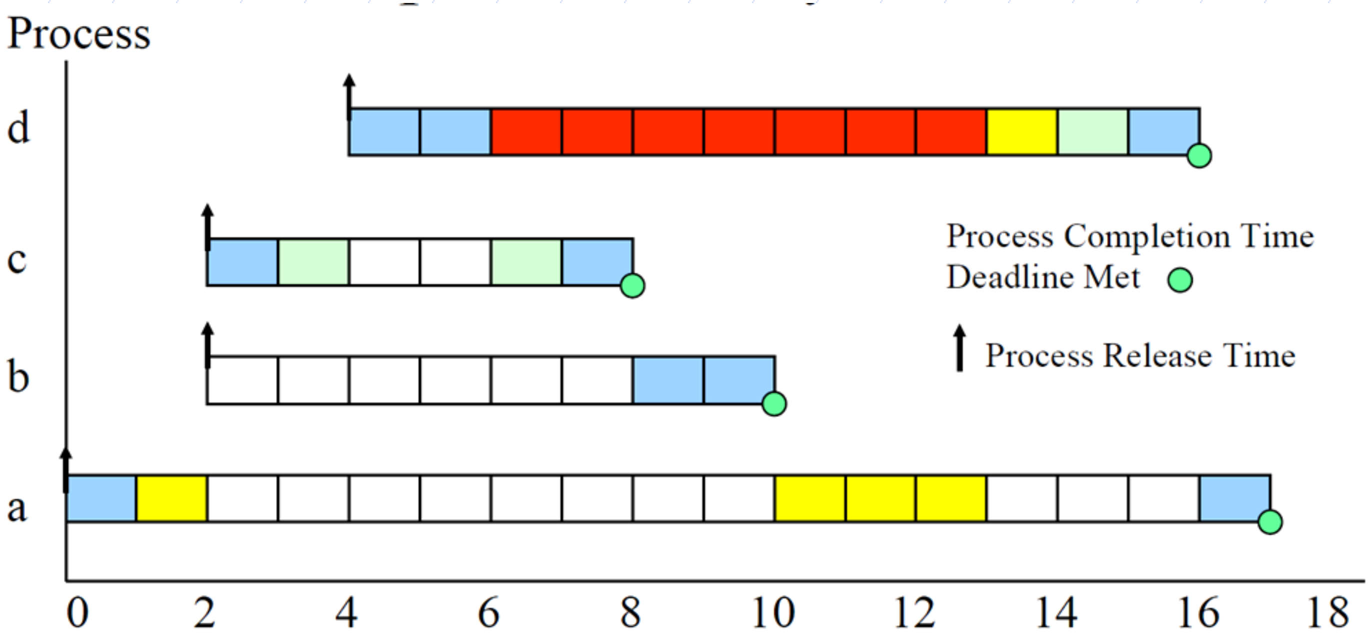
Priority Inversion….

Priority Inversion

Process	Priority	Execution Sequence	Release Time
a	1	EQQQE	0
b	2	EE	2
c	3	EVVE	2
d	4	EEQVE	4

- **E** = execution on CPU
- **Q** and **V** = execution on resource
- Higher the number = higher the priority

Priority Inversion



Executing

Preempted

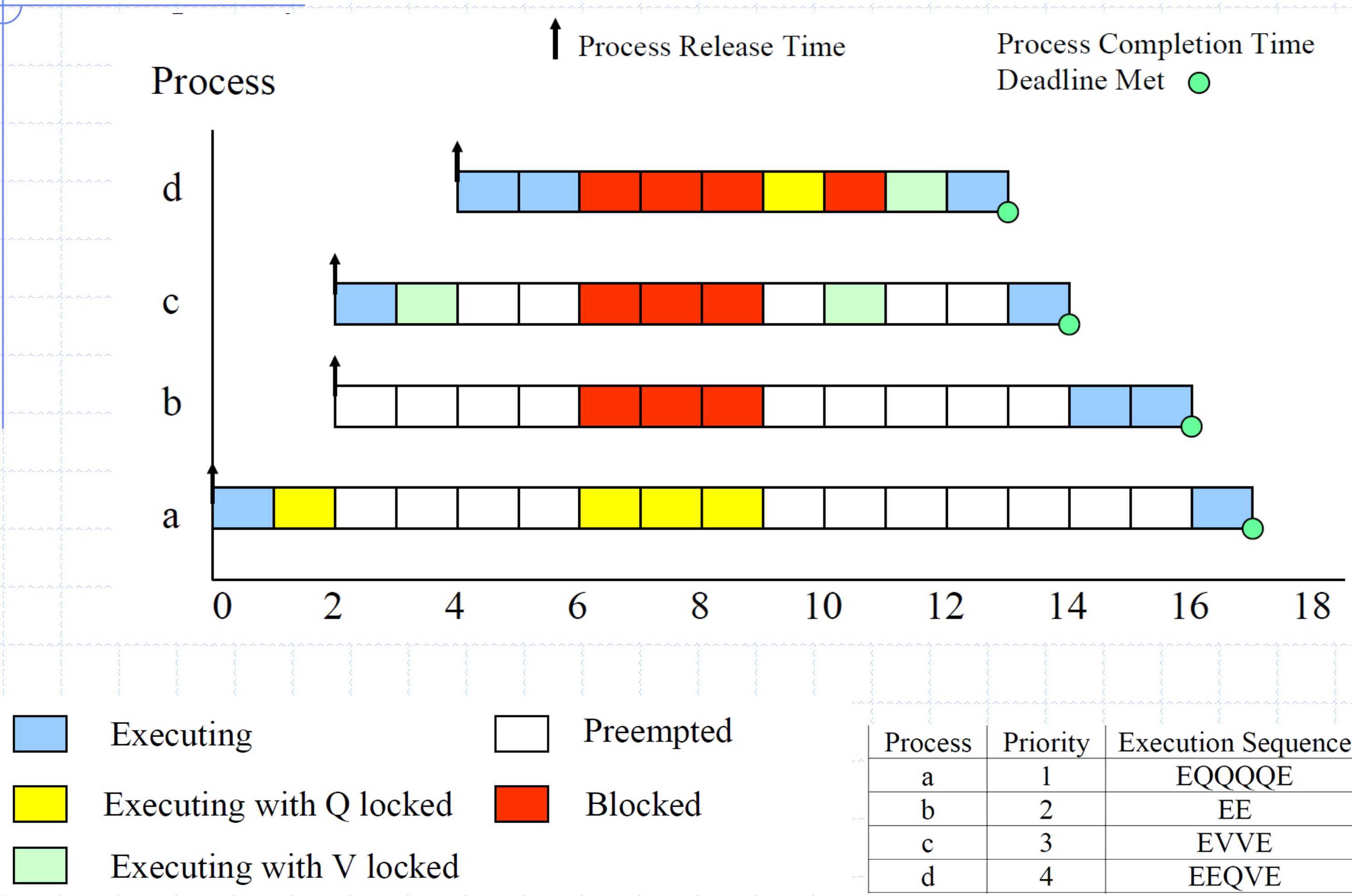
Executing with Q locked

Blocked

Executing with V locked

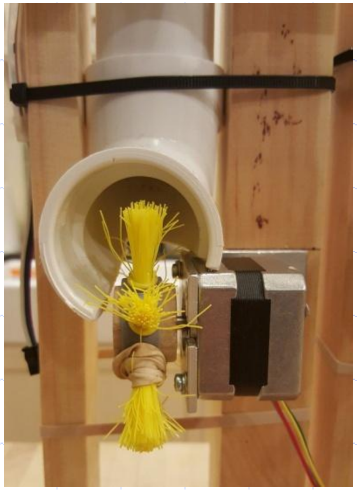
Process	Priority	Execution Sequence	Release Time
a	1	EQQQQE	0
b	2	EE	2
c	3	EVVE	2
d	4	EEQVE	4

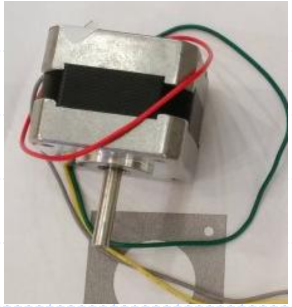
Priority Inversion Solution

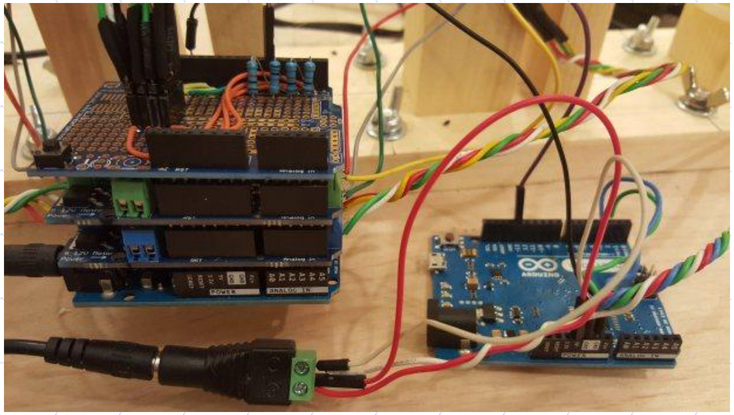


Process	Priority	Execution Sequence	Release Time
a	1	EQQQQE	0
b	2	EE	2
c	3	EVVE	2
d	4	EEQVE	4

Halloween Candy Dispenser







http://makezine.com/projects/build-an-arduino-powered-candy-vending-machine/