Intelligent Signal Processing

Multimedia Process Scheduling

Angelo Ciaramella

Homogeneous processes

- The simplest kind of video server
  - support the display of a fixed number of movies
    - same frame rate, video resolution, data rate, and other parameters
  - For each movie, there is a single process (or thread)
- NTSC 30 times per second
  - number of processes is small enough that all the work can be done in one frame time
    - round-robin scheduling
    - this model is rarely applicable in reality

Real-time scheduling

- Real applications
  - the number of users changes as viewers come and go
  - frame sizes vary wildly due to the nature of video compression
  - different movies may have different resolutions
  - multiple processes competing for the CPU

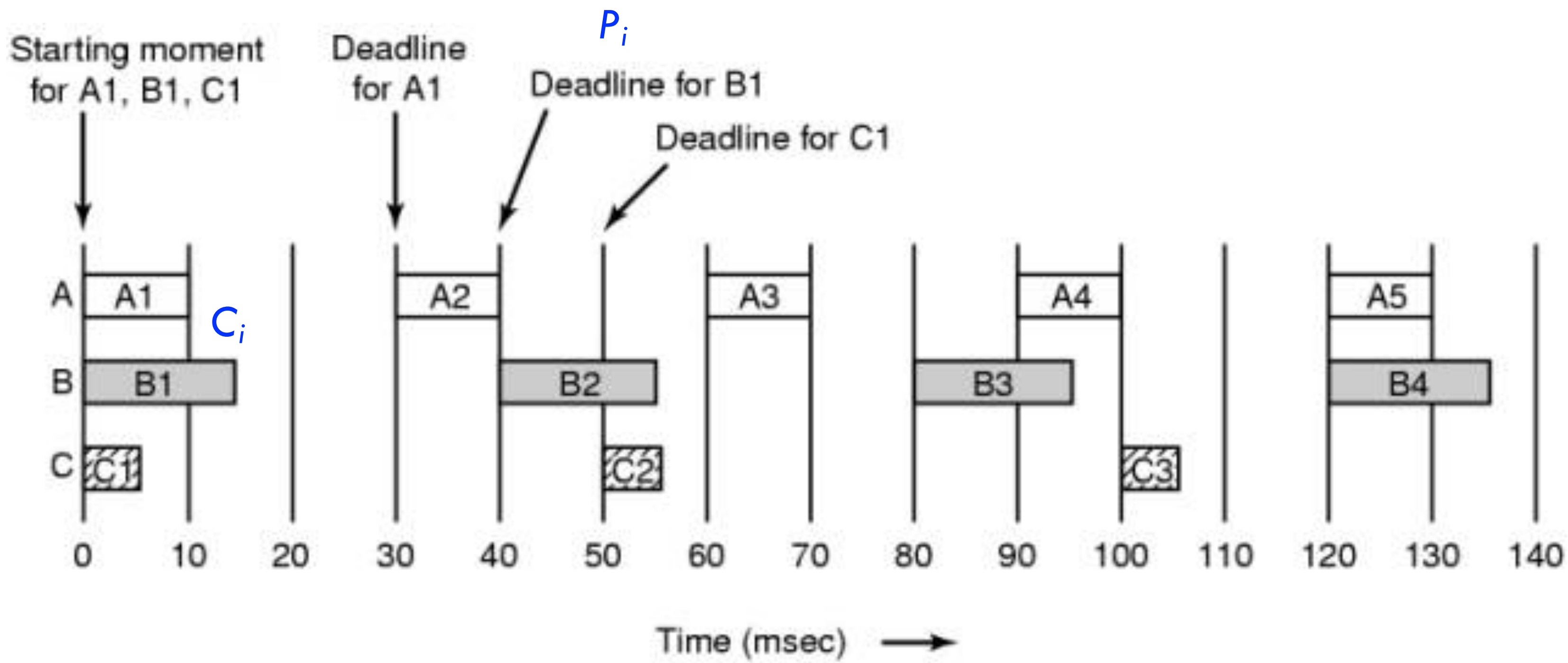
- Real time scheduling
  - the system knows the frequency at which each process must run
  - how much work it has to do
  - what its next deadline is

Real time-scheduling

Process	*Priodicity*	*CPU time*
A	33 Hz (NTSC)	10 ms
B	25 Hz (PAL)	15 ms
C	20 Hz (PAL slow connection)	5 ms

Example of processes

Real time-scheduling



Three periodic processes, each displaying a movie. The frame rates and processing requirements per frame are different for each movie.

Schedulable processes

- Schedulable condition
  - if process *i* has period P_i msec and requires C_i msec of CPU time per frame, the system is schedulable if and only if

$$\sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1$$

Schedulable processes

Process	Ci/Pi
A	10/30
B	15/40
C	5/50

The system of processes is schedulable since the total is 0.808 of the CPU

Real-time algorithms

- Real-time algorithms can be
  - static
    - assign each process a fixed priority in advance and then do prioritized preemptive scheduling using those priorities
  - dynamic
    - does not have fixed priorities

Rate Monotonic Scheduling

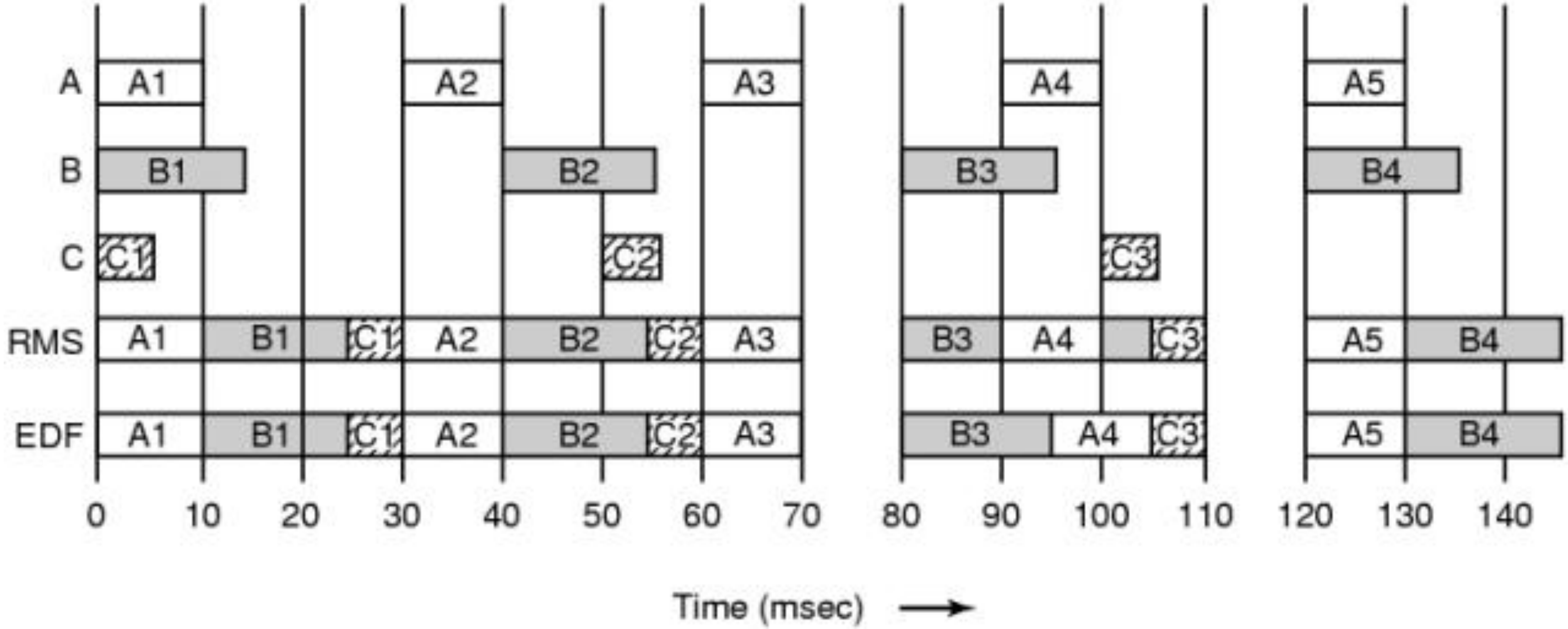
- Rate Monotonic Scheduling (RMS)
  - Liu and Layland, 1973
  - real-time scheduling algorithm for preemptable, periodic processes
- Conditions
  - each periodic process must complete within its period
  - no process is dependent on any other process
  - each process needs the same amount of CPU time on each burst
  - any nonperiodic processes have no deadlines
  - process preemption occurs instantaneously and with no overhead

Rate Monotonic Scheduling

- Rate Monotonic Scheduling
  - works by assigning each process a fixed priority equal to the frequency of occurrence of its triggering event
  - Liu and Layland proved that RMS is optimal among the class of static scheduling algorithms

Process	Priority
A	33
B	25
C	20

RMS



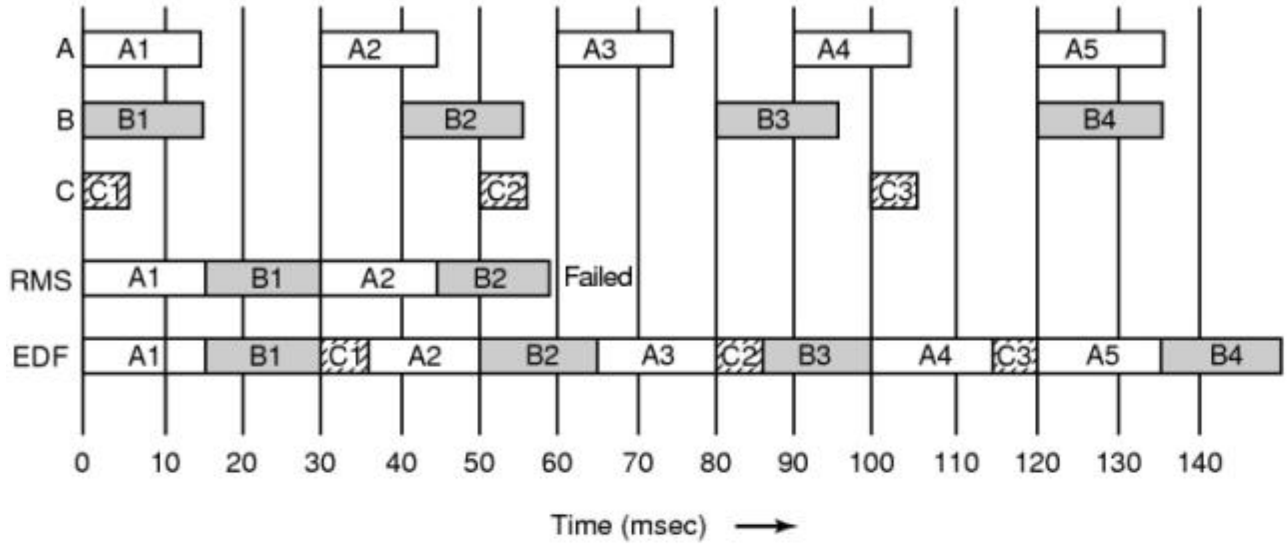
An example of RMS and EDF real-time scheduling

Earliest Deadline First Scheduling

- Earliest Deadline First Scheduling (EDF)
  - dynamic algorithm that does not require processes to be periodic

- Algorithm
  - a process needs CPU time, it announces its presence and its deadline
  - the scheduler keeps a list of runnable processes, *sorted on deadline*
  - the algorithm runs the first process on the list, the one with the closest deadline
  - whenever a new process becomes ready, the system checks to see if its deadline occurs before that of the currently running process
    - If so, the new process preempts the current one

RMS vs EDF



Example of RMS and EDF real-time scheduling (schedulable processes)

RMS

- Any system of periodic processes, if

$$\sum_{i=1}^{m} \frac{C_i}{P_i} \leq m\left(2^{1/m} - 1\right)$$

then RMS is guaranteed to work

RMS

# of processes	*CPU utilization*
3	0.780
4	0.757
5	0.743
10	0.718
20	0.705
100	0.696
infinity	*ln 2*

CPU utilization by using RMS

EDF

- EDF
  - always works for any schedulable set of processes
  - it can achieve 100% CPU utilization
  - the price paid is a more complex algorithm