Lecture 11 – Real-Time Scheduling
Networking Background

PORTLAND STATE UNIVERSITY

Admin

- PA2
  - About that deadline… ~~Wed. Nov. 18, 2015~~

FRI MIDNIGHT SOURCE
ELECTRONIC
MON. MORNING WRITE-UP
↳ CHECK FAQ

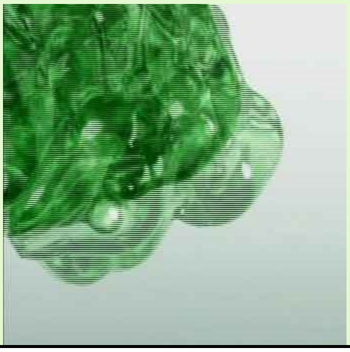
- HW 5
  - On website
  - Due Mon. Nov. 23, 2015

PORTLAND STATE UNIVERSITY

Interlacing issues

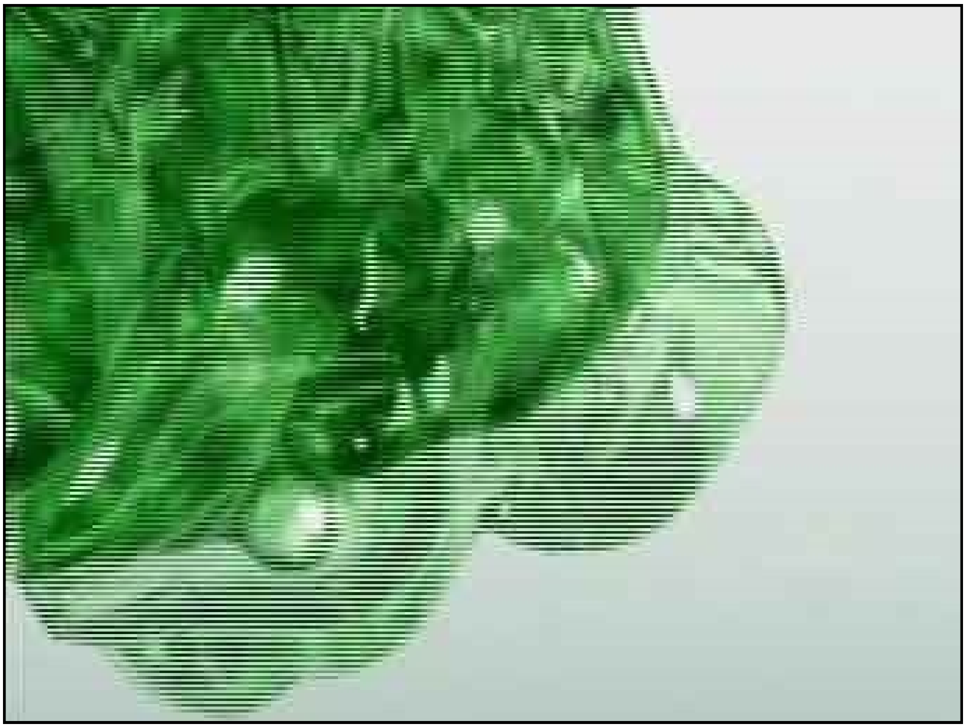
FIELD 1

FIELD 2

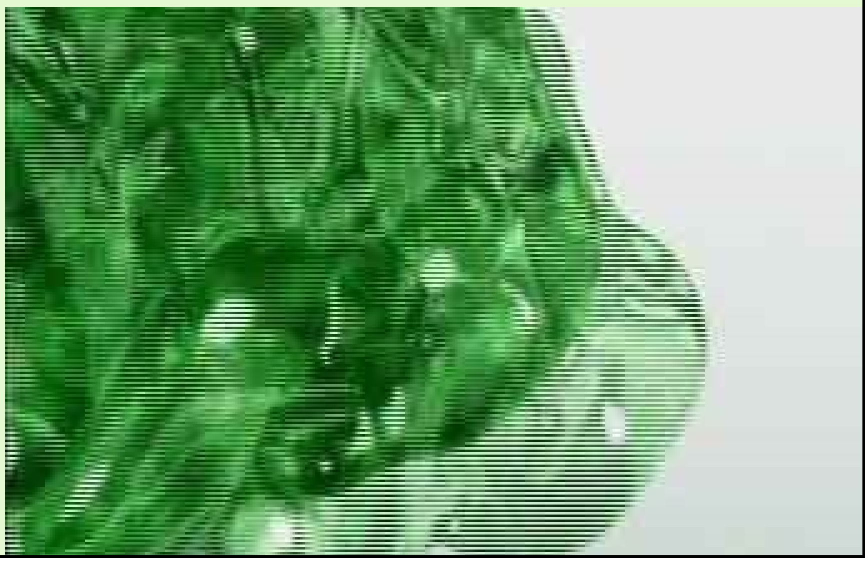


www.100fps.com

PORTLAND STATE UNIVERSITY

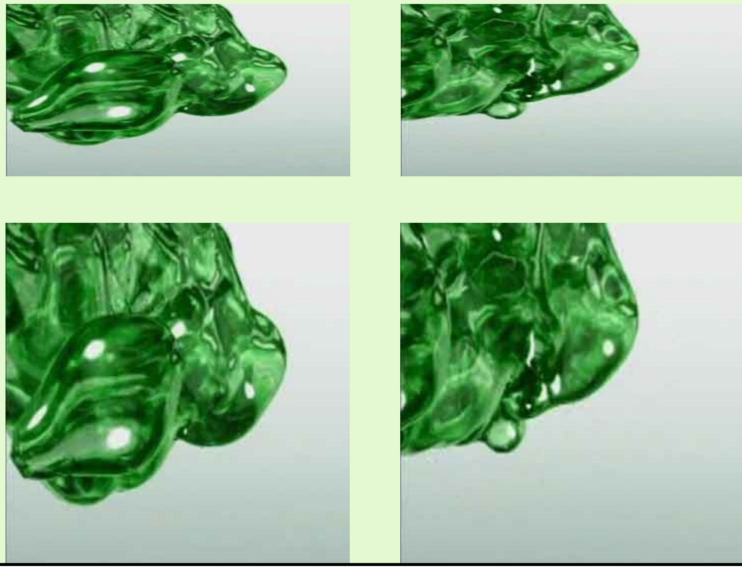


De-interlacing via weave



De-interlacing via blending

“De-interlacing” BoB

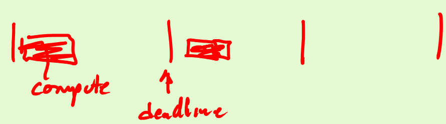


RT Scheduler Types

Assumptions

Hard deadlines are periodic

Tasks complete before next request occurs



Tasks are independent

Run-time is constant for each task

Two types of schedulers:

Static - fixed priorities - rate monotonic scheduling

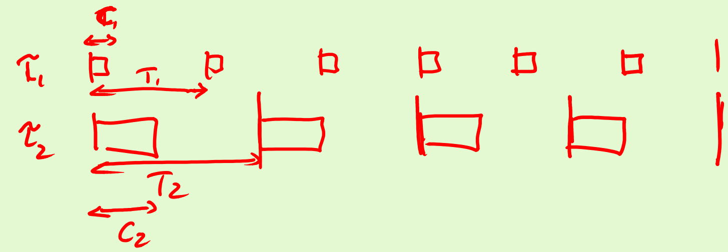
dynamic priority - deadline driven scheduling

Terminology

$\tau_1, \tau_2 \ldots \tau_m \Rightarrow m$ periodic tasks

$T_1 \ldots T_m \Rightarrow$ request periods

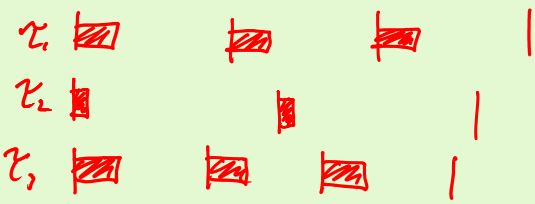
$C_1 \ldots C_m \Rightarrow$ run time of each task



Critical Instants / Time Zones

Critical instant when a task is requested with all higher priority tasks

time zone - time allocated to complete task



Using critical instant $\Rightarrow$ determine <u>feasibility</u> of the task set

Priority assignment can determine the feasibility & important to pick right!

Priorities and Feasibility

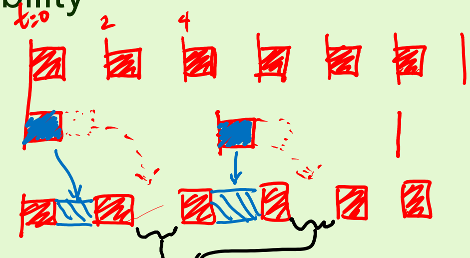
Assuming fixed priority scheduling

$T_1 = 2 \quad C_1 = 1$

$T_2 = 5 \quad C_2 = 1$

Priority $\tau_1 > \tau_2$

run time



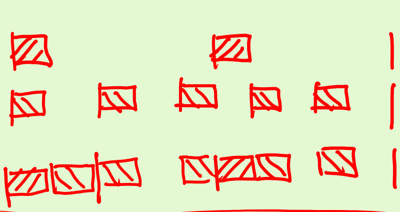
$T_2 = 5 \quad C_2 = 2$ also works

priority $\tau_2 > \tau_1$?

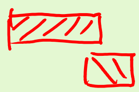
$T_2 = 5 \quad C_2 = 1$

$T_1 = 2 \quad C_1 = 1$

run time



Can't have $C_2 = 2$ τ_2



τ_1 would miss deadline

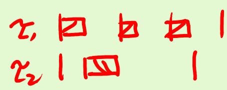
Portland State University

Priorities and Feasibility

These examples help us understand priority assignment

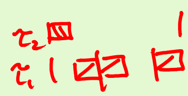
From example 1 $prior(\tau_1) > prior(\tau_2)$

$$\left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + C_2 \le T_2 \quad (1)$$



Example 2

$$C_1 + C_2 <= T_1 \quad (2)$$



multiply (2) by $\left\lfloor \frac{T_2}{T_1} \right\rfloor$

$$\left\lfloor \frac{T_2}{T_1} \right\rfloor C_1 + \left\lfloor \frac{T_2}{T_1} \right\rfloor C_2 \le \left\lfloor \frac{T_2}{T_1} \right\rfloor T_1 <= T_2$$

$$\frac{T_2}{T_1} C_1 + \frac{T_2}{T_1} C_2 \le \cancel{T_1} \frac{T_2}{\cancel{T_1}}$$

(2) implies (1) but (1) does not imply (2)

~~Somethin~~

Portland State University

Priorities and Feasibility

Suppose have τ_1 & τ_2 with $T_1 < T_2$

HP T_2

T_1

If $prior(\tau_2) > prior(\tau_1)$ is feasible

$priority(\tau_1) > prior(\tau_2)$ is also feasible

Suppose τ_2 & τ_1 w/ $T_1 < T_2$

If $prior(\tau_1) > prior(\tau_2)$ is feasible

HP T_1

$prior(\tau_2) > prior(\tau_1)$ is NOT necessarily feasible

T_2

If we are scheduling R.T. tasks w/ fixed priority always run smaller periods w/ higher priority

Rate Monotonic Scheduling (RMS)

↳ Fixed priority preemptive scheduler

└ you interrupt lower priority

Assign higher rate to higher priority processes

RMS is optimal in that no other assignment rule can schedule a task set which cannot be scheduled by RMS priority

Determine feasibility?

$$U_{proc} = \sum_{i=1}^{m} \left(\frac{C_i}{T_i}\right)$$

processor utilization

Do not memorize

$$U_{theoretical} = m\left(2^{\left(\frac{1}{m}\right)} - 1\right)$$

Most use 69 or 70% for $U_{theoretical}$

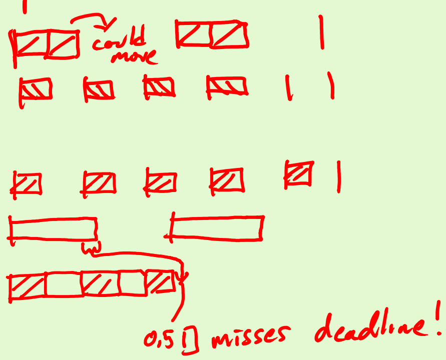
If ($U_{proc} < U_{theoretical}$) it's feasible

else more expensive test needed

Deadline Driven Scheduling

RMS is a rigid but simple scheduler

Observation previous



Another example:

$T_1 = 2 \quad c_1 = 1$

$T_2 = 5 \quad C_2 = 2.5$

We need a scheme that
ensures that everything can run
w/o missing deadlines → dynamic priority

Deadline Driven Scheduling

Dynamic priority preemptive scheduler

Priorities are assigned to tasks according
to deadline of current request

Important results:

Theorem: When deadline scheduling is used,
there is no processor idle time prior to
an over flow (missed deadline)

Theorem: for m tasks using deadline scheduling
is feasible if and only if

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

Referred to as "EDF"

Sounds great but
deadlines have to be
continually monitored.

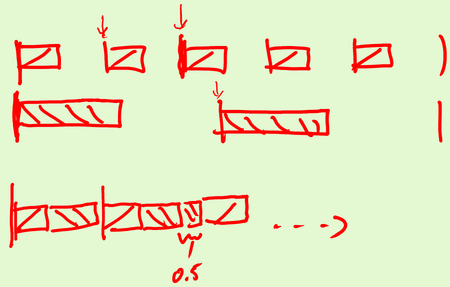
RMS vs. EDF

Example from slide -2

$T_1 = 2 \quad C_1 = 1$

$T_2 = 5 \quad C_2 = 2.5$

Under EDF



Real-time Assumptions Revisited

Assumptions

MM

Deadlines periodic ✓ but soft deadlines.

Tasks independent mostly true

audio/video need to be synched

Run-time constant for each task not true

I-frame vs. P-frame

Motion vectors + residual

vs.

just motion vector

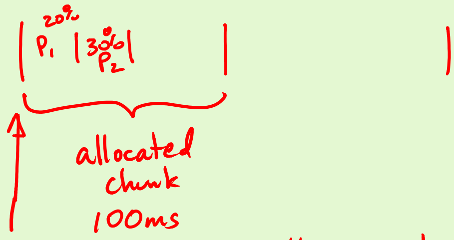
vs.

skipped block

Another Way to Schedule

Proportional Share Scheduling

Every process gets a share of the O.S/ resources



Give all processes allowance to use in this period. When allowance gone must wait until next or until processor is free