x(t)	X(s)	X(z)
$\delta(t)=\begin{cases}1 & t=0,\\ 0 & t=kT,\ k\neq 0\end{cases}$	1	1
$\delta(t-kT)=\begin{cases}1 & t=kT,\\ 0 & t\neq kT\end{cases}$	e^{-kTs}	z^{-k}
$u(t)$, unit step	$\frac{1}{s}$	$\frac{z}{z-1}$
t	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$
t^2	$\frac{2}{s^3}$	$\frac{T^2z(z+1)}{(z-1)^3}$
e^{-at}	$\frac{1}{s+a}$	$\frac{z}{z-e^{-aT}}$
$1-e^{-at}$	$\frac{a}{s(s+a)}$	$\frac{(1-e^{-aT})z}{(z-1)(z-e^{-aT})}$
te^{-at}	$\frac{1}{(s+a)^2}$	$\frac{Tze^{-aT}}{(z-e^{-aT})^2}$
t^2e^{-at}	$\frac{2}{(s+a)^3}$	$\frac{T^2e^{-aT}z(z+e^{-aT})}{(z-e^{-aT})^3}$
$be^{-bt}-ae^{-at}$	$\frac{(b-a)s}{(s+a)(s+b)}$	$\frac{z[z(b-a)-(be^{-aT}-ae^{-bT})]}{(z-e^{-aT})(z-e^{-bT})}$
$\sin\omega t$	$\frac{\omega}{s^2+\omega^2}$	$\frac{z\sin\omega T}{z^2-2z\cos\omega T+1}$
$\cos\omega t$	$\frac{s}{s^2+\omega^2}$	$\frac{z(z-\cos\omega T)}{z^2-2z\cos\omega T+1}$
$e^{-at}\sin\omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$	$\frac{(ze^{-aT}\sin\omega T)}{z^2-2ze^{-aT}\cos\omega T+e^{-2aT}}$

derivatives and integral

if $\mathcal{L}[f(t)] = F(s)$ then we have

$$\mathcal{L}[f'(t)] = sF(s) - f(0)$$
$$\mathcal{L}[\int_0^t f(t)dt] = \frac{F(s)}{s}$$

For higher derivatives we have $\mathcal{L}[f''(t)] = s^2F(s) - sf(0) - f'(0)$

inverse form

$$\mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j}\lim_{\omega\to\infty}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{st}\,ds$$

poles and zeros

zeros and poles generate the amplitude for both forced and natural res

$$Y(s) = \frac{s+2}{s(s+5)}$$

$s = 0, -5$ are poles and $s = -2$ are zeros

poles

- at origin, generated step function
- at -5 generate transient response e^{-5t}

thus the time domain output is $y(t) = 1 - e^{-at}$

time constant

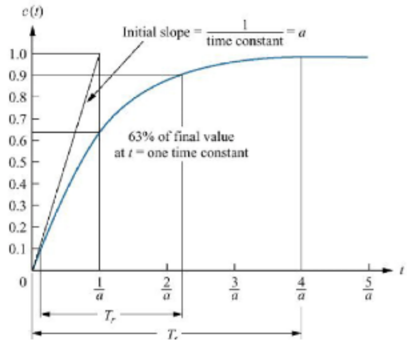
usually, $t = \frac{1}{a}$, and $y(t) = 0.63$, hence 63.2% to find the rise time.

$w_n = \sqrt{b}$ is the frequency of oscillation of this system.

settling time T_s

T_s, time for response to reach and stay with 2% of its final value

for first order: $T_s = \frac{4}{a}$



peak time T_p

time required to reach the first or maximum peak

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$$

damping ratio is defined as:

$$\zeta = \frac{\text{exponential decay frequency}}{\text{natural frequency}} = \frac{|\sigma|}{w_n}$$

So that $a = 2\zeta w_n$

%OS (percent overshoot)

$$\%OS = e^{\zeta\pi/\sqrt{1-\zeta^2}} \times 100\%$$

or in terms of damping ratio ζ:

$$\zeta = \frac{-\ln\frac{\%OS}{100}}{\sqrt{\pi^2 + \ln^2(\frac{\%OS}{100})}}$$

second-order systems

general order system:

$$G(s) = \frac{b}{s^2 + as + b}$$

Thus the pole for this system:

$$s_1, s_2 = \frac{-a \pm \sqrt{a^2 - 4b}}{2}$$

$$G(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

$$s_1, s_2 = -\zeta w_n \pm w_n\sqrt{\zeta^2 - 1}$$

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$$

$$T_s \cong \frac{4}{\zeta\omega_n}$$

general second order

$$G(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

$$s_1, s_2 = -\zeta w_n \pm w_n\sqrt{\zeta^2 - 1}$$

Condition	Poles	pole type	Damping Ratio (ζ)	Natural Response $c(t)$
Undamped	$\pm j\omega_n$	imaginary	$\zeta = 0$	$A\cos(\omega_n t - \varphi)$
Underdamped	$\omega_d \pm j\omega_d$	complex	$0 < \zeta < 1$	$Ae^{(-\sigma_d)t}\cos(\omega_d t - \varphi)$ where $w_d = w_n\sqrt{1-\zeta^2}$
critically damped	σ_1	real	$\zeta = 1$	$Kte^{\sigma_1 t}$
overdamped	$\sigma_1 \quad \sigma_2$	real	$\zeta > 1$	$K(e^{\sigma_1 t} + e^{\sigma_2 t})$

underdamped second-order step response

Transfer function $C(s)$ is given by

$$C(s) = \frac{w_n^2}{s(s^2 + 2\zeta w_n s + w_n^2)}$$

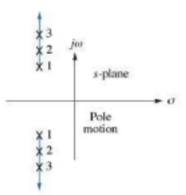
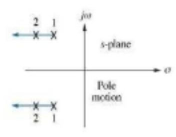
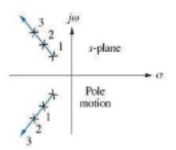
response in time-domain via inverse Laplace transform:

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}}e^{-\zeta w_n t}\cos(\sqrt{1-\zeta^2}w_n t + \varphi)$$

where $\varphi = \tan^{-1}(\frac{\zeta}{\sqrt{1-\zeta^2}})$

Transfer function of Zero-Order hold

$$\mathcal{L}(u(t) - u(t-T)) = \frac{1}{s} - \frac{e^{sT}}{s}$$

location of poles	response
Same envelope	
Same frequency	
Same overshoot	

T is the sampling period, and $\frac{1}{T}$ is the sampling rate in *cycles per second*

$$\text{error} = \frac{M}{2^{n-1}}$$

where n is number of bits used for digitalisatic

resolution of A/D converter

minimum value of the output that can be re number, or $\frac{M}{2^n}$

sampled data system

reference input r is the sequence of sample values $r(kT)$

A sampler is a switch that closes every T seconds:

$$r^*(t) = \sum_{k=0}^{\infty} r(kT)\delta(t - kT) \quad (t > 0)$$

Transfer function of sampled data:

$$R^*(s) = \mathcal{L}(r^*(t)) = \sum_{k=0}^{\infty} r(kT)e^{-ksT}$$

finding the discrete transfer function

$G(s) = \frac{s^2+4s+3}{s^3+6s^2+8s}$

$$G(s) = \frac{s^2+4s+3}{s^3+6s^2+8s} = \frac{0.375}{s} + \frac{0.25}{s+2} + \frac{0.375}{s+4}$$
$$G(t) = \mathcal{L}^{-1}(G(s)) = 0.375 + 0.25e^{-2t} + 0.375e^{-4t}$$
$$G(z) = Z(G(t)) = 0.375\frac{z}{z-1} + 0.25\frac{z}{z-e^{-2T}} + 0.375\frac{z}{z-e^{-4T}}$$

Let $z = e^{sT}$, we have the following definition:

z-transform

$$Z\{r(t)\} = F(z) = Z(r^*(t)) = \sum_{k=0}^{\infty} r(kT)z^{-k}$$

mapping from s-plane to z-plane

$$z = e^{\alpha T}(\cos\omega T + j\sin\omega T)$$

we assume $s = \alpha - j\omega$

Location on s-plane	Value of α	Value of $e^{\alpha T}$	Mapping on z-plane
Imaginary axis ($j\omega$)	$\alpha = 0$	$e^{\alpha T} = 1$	On unit circle
Right half-plane	$\alpha > 0$	$e^{\alpha T} > 1$	Outside unit circle
Left half-plane	$\alpha < 0$	$e^{\alpha T} < 1$	Inside unit circle

stability

system	pole location criteria on z-plane
Stable	All poles inside unit circle
Unstable	Any poles outside unit circle
Marginally Stable	One or more poles on unit circle, remaining poles inside unit circle

final value theorem

definition

If $\lim_{k\to\infty} x(k)$ exists, then the follow exists:

$$\lim_{k\to\infty} x(k) = \lim_{z\to 1}(z-1)X(z)$$

Linearity: if $x(n) = af_1(n) + bf_2(n)$ then $X(z) = aF_1(z) + bF_2(z)$

Time shifting:

$$Z[x(t)] = X(z)$$

$$Z[x(n+k)] = z^kX(z) - z^k\sum_{i=0}^{k-1} x(i)z^{-i}$$

$$Z[x(n-k)] = z^{-k}X(z) + z^{-k}\sum_{i=0}^{k-1} x(-i)z^{i}$$

PID control

$$G_C(s) = K_p + \frac{K_I}{s} + K_D s$$

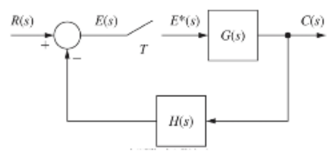
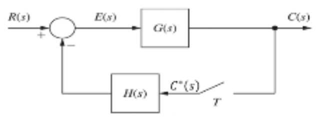
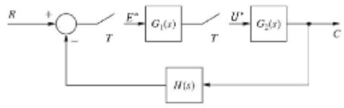
in time domain:

$$u(t) = K_P e(t) + K_I \int_0^t e(\eta) d\eta + K_D \frac{d(e(t))}{dt}$$

Proportional (P)	$\frac{K_p G_p}{1+K_p G_p}$	Basic control action	- Affects speed of response - Cannot eliminate steady-state error
Integral (I)	$\frac{K_I}{s^2+s+K_I}$	Integrates error over time	- Eliminates steady-state error - Output reaches 1 at steady state
PI	$\frac{K_I+sK_p}{s^2+(1+K_p)s+K_I}$	Combines P and I	- P impacts response speed - I forces zero steady-state error
Derivative (D)	$\frac{K_D s}{(1+K_D)s+1}$	Based on rate of change	- Adds open-loop zero - C... ct stability ... damping

Component	Discrete-Time Equation
Proportional	$u(k) = K_P e(k)$
Integral	$u(k) = K_I T \sum_{i=1}^{k} e(i)$
Derivative	$u(k) = \frac{K_D}{T}[e(k) - e(k-1)]$

approximate of PID controller:
$u(k) = K_P e(k) + K_I T \sum_{i=1}^{n} e(i) + \frac{K_D}{T}[e(k) - e(k-1)]$

System Type	Transfer Function	Diagram
Basis	$\frac{C(z)}{R(z)} = \frac{G(z)}{1+Z[G(s)H(s)]}$	
w/ digital sensing device	$C(z) = \frac{Z[G(s)R(s)]}{1+Z(G(s)H(s))}$	
w/ digital controller	$\frac{C(z)}{R(z)} = \frac{G_1(z)G_1(z)}{1+G_1(z)Z(G_2(s)H(s))}$	

Assume the transfer function is given by

$$D(s) = \frac{U(s)}{E(s)} = K_0 \frac{s+a}{s+b}$$

difference equation

$$u(k) = (1-bT)u(k-1) + K_0(aT-1)E(k-1) + K_0 e(k)$$

The corresponding z-transform

$$\frac{U(z)}{E(z)} = \frac{K_0(aT-1)z^{-1} + K_0}{1+(bT-1)z^{-1}} = \frac{K_0 z + K_0(aT-1)}{z+(bT-1)}$$
$$= [K_0(aT-1) + zK_0]/[z+(bT-1)]$$

$D(s)$	rule	z-transfer function $D(z)$	approximation	z-plane to s-plane	stability
$\frac{a}{s+a}$	forward	$\frac{a}{(z-1)/T+a}$	$s \leftarrow \frac{z-1}{T}$	$z \leftarrow sT+1$	discrete → continuou
$\frac{a}{s+a}$	backward	$\frac{a}{(z-1)/(Tz)+a}$	$s \leftarrow \frac{z-1}{Tz}$	$z \leftarrow \frac{1}{1-Ts}$	discrete ← continuou
$\frac{a}{s+a}$	trapzoid	$\frac{a}{(2/T)[(z-1)/(z+1)]+a}$	$s \leftarrow \frac{2}{T}\frac{z-1}{z+1}$	$z \leftarrow \frac{1+Ts/2}{1-Ts/2}$	discrete ↔ continuou

z-transform of difference equation

example: Given $D(s) = \frac{a}{s+a}$, $u(kT) = u(k)$

$U(s)(s+a) = aE(s)$ (Laplace transform) gives
$\frac{u(k+1)-u(k)}{T} + au(k) = ae(k)$

difference equation is $u(k+1) = (1-aT)u(k) + aTe(k)$

z-transform is $\frac{U(z)}{E(z)} = \frac{aTz^{-1}}{1+(aT-1)z^{-1}} = \frac{aT}{z+(aT-1)}$

discrete equivalent

Consider the example

$$D(s) = \frac{U(s)}{E(s)} = \frac{a}{s+a} \to U(s)s = aE(s) - aU(s)$$
$$\to u'(t) = -au(t) + ae(t)$$

$$u(t) = \int_0^t [-au(\tau) + ae(\tau)] d\tau$$

discrete system

$$u(kT) = u(kT-T) + \int_{kT-T}^{kT} [-au(\tau) + ae(\tau)] d\tau$$

Steps to plot closed-loop poles

1. Derive the open loop function $K\overline{GH}$.
2. Factor numerator and denominator to get open loop zeros and p
3. Plot roots of $1 + K\overline{GH} = 0$ in z-Plane as K varies.

1. Loci originate on the poles of $K\overline{GH}$ and terminate on its zeros.
2. The loci are symmetrical with respect to the real axis.
3. The number of asymptotes is equal to the number of poles of $K\overline{GH}$, n_p, minus the number of its zeros, n_z. The angles of the asymptotes are found by $\theta_a = \frac{(2k+1)\pi}{n_p - n_z}$, $k = 0, 1, 2, \ldots (n_p - n_z - 1)$, where n_p is # of finite poles and n_z is # of finite zeros.
4. The origin of the asymptotes on the real axis is given by

$$\sigma = \frac{\sum \text{poles of } \overline{GH}(z) - \sum \text{zeros of } \overline{GH}(z)}{n_- - n_-}$$

5. The breakaway point for the locus between two poles (or the break-in point for the locus between two zeros) is found by

$$\frac{d[\overline{GH}(z)]}{dz} = 0$$

Given the following frequency domain function: $F(s) = \frac{2s+a}{s(s-a)}$
ustify whether the Final Value Theorem can or cannot be used to find he steady state value of $f(t)$.

Answer: Not applicable, since the steady state tends to infinity.
$\lim f(t) = \lim (3e^{at} - 1) = \infty$

Solve $y[k+2] - 5y[k+1] + 6y[k] = 0$, where $y[0] = 0$, $y[1] = 2$.

$$\mathcal{Z}\{y[k+2]\} - 5\mathcal{Z}\{y[k+1]\} + 6\mathcal{Z}\{y[k]\} = 0$$

Taking z transforms:

$$z^2Y(z) - zy[0] - zy[1] - 5zY(z) + 5zy[0] + 6Y(z) = 0$$

Rearranging and using initial conditions:

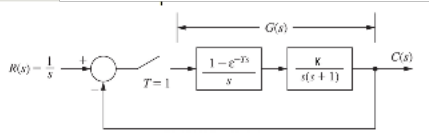
$$(z^2 - 5z + 6)Y(z) = 2z$$
$$Y(z) = \frac{2z}{z^2 - 5z + 6}$$

Using partial fractions:

$$Y(z) = \frac{2}{z-3} + \frac{2}{z-2}$$

Using inverse transforms straight from the table to get the solution:

$$y[k] = 2 \times 3^k - 2 \times 2^k$$



- Consider the system, we have

$$G(z) = (1 - z^{-1}) Z\left[\frac{K}{s^2(s+1)}\right]_{(T=1)} = K\frac{z-1}{z}\left[\frac{z^2e^{-1} + z - 2ze^{-1}}{(z-1)^2(z-e^{-1})}\right] = \frac{K(ze^{-1} + 1 - 2e^{-1})}{(z-1)(z-e^{-1})}$$

- Hence, the open loop transfer function $G(z) = \frac{0.368K(z+0.718)}{(z-1)(z-0.368)}$

correctness: $|C(t) - Cs(t)| < \epsilon$

`fork()`

- create a `child` process that is identical to its parents, return `0` to child process and pid
- add a lot of overhead as duplicated. **Data space is not shared**

drift is RoC of the clock value from perfect clock. Given clock has bounded drift ρ

$$| \frac{dC(t)}{dt} - 1 | < \rho$$

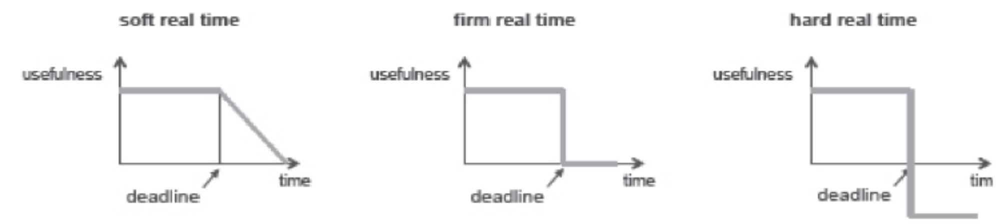
	OS	RTOS
philos	time-sharing	event-driven
requirements	high-throughput	schedulablity (meet all hard deadlines)
metrics	fast avg-response	ensureed worst-case response
overload	fairness	meet critical deadlines

Monotonicity: $\forall t_2 > t_1 : C(t_2) > C(t_1)$

preemption && `syscall`

> The act of temporarily interrupting a currently scheduled task for higher priority tasks.

> NOTE: `make` doesn't recompile if DAG is not changed.



process

- independent execution, logical unit of work scheduled by OS
- in virtual memory:
  - Stack: store local variables and function arguments
  - Heaps: dyn located (think of `malloc`, `calloc`)
  - BSS segment: uninit data
  - Data segment: init data (global & static variables)
  - text: RO region containing program instructions

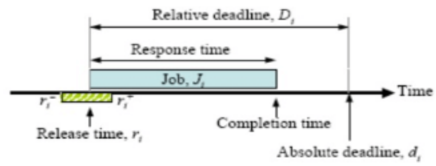
threads

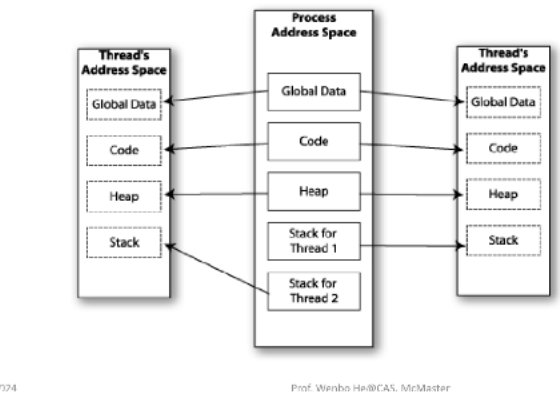
- program-wide resources: global data & instruction
- execution state of control stream
- shared address space for faster context switching

- Needs synchronisation (global variables are shared between threads)
- lack robustness (one thread can crash the whole program)

	interrupt	polling
speed	fast	slow
efficiency	good	poor
cpu-waste	low	high
multitasking	yes	yes
complexity	high	low
debug	difficult	easy

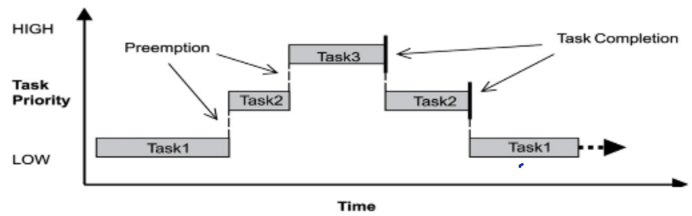
	stack	heap
creation	`Member m`	`Member*m = new Member()`
lifetime	function runs to completion	delete, free is called
grow	fixed	dyn added by OS
err	stack overflow	heap fragmentation
when	size of memory is known, data size is small	large scale dyn mem

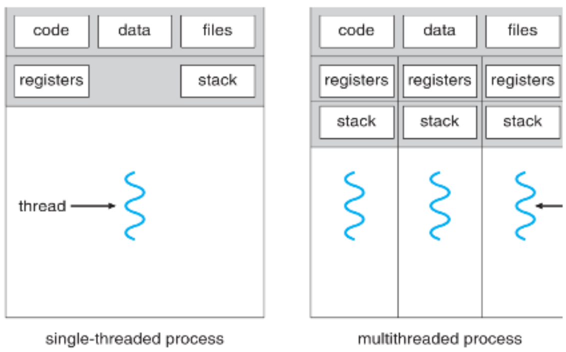




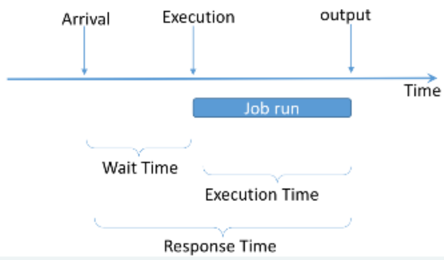
scheduling

1. Priority-based preemptive scheduling





Temporal parameters:



Let the following be the scheduling parameters:

desc	var
# of tasks	n
release/arrival-time	$r_{i,j}$
absolute deadline	d_i
relative deadline	$D_i = r_{i,j} - d_i$
execution time	e_i
response time	R_i

Utilisation factor u_i

for a task T_i with execution time e_i and period p_i is given by

$$u_i = \frac{e_i}{p_i}$$

For system with n tasks overall system utilisation is $U = \sum_{i=1}^{n} u_i$

Period p_i of a periodic task T_i is **min length** of all time intervales between release times of consecutive tasks.

Phase of a Task ϕ_i is the release time $r_{i,1}$ of a task T_i, or $\phi_i = r_{i,1}$

in phase are first instances of several tasks that are released simultaneously

Representation

a periodic task T_i can be represented by:

- 4-tuple (ϕ_i, P_i, e_i, D_i)
- 3-tuple (P_i, e_i, D_i), or $(0, P_i, e_i, D_i)$
- 2-tuple (P_i, e_i), or $(0, P_i, e_i, P_i)$

cyclic executive

assume tasks are non-preemptive, jobs parameters with hard deadlines known.

- no race condition, no deadlock, just function call
- however, very brittle, number of frame F can be large, release times of tasks must be fixed

hyperperiod

> is the least common multiple (lcm)

maximum num of arriving jobs

$$N = \sum_{i=1}^{n} \frac{H}{p_i}$$

Frames: each task must fit within a single frame with size f => number of frames $F = \frac{H}{f}$

C1: A job must fit in a frame, or $f \geq \max e_i \forall 1 \leq i \leq n$ for all tasks

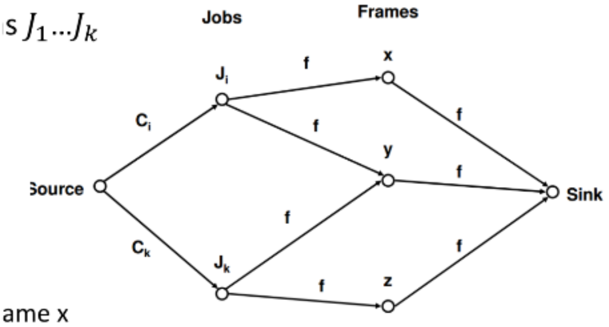
C2: hyperperiod has an integer number of frames, or $\frac{H}{f} =$ integer

deadline-monotonic

C3: $2f - \gcd(P_i, f) \leq D_i$ per task.

- if every task has period equal to relative deadline, same as RM
- arbitrary deadlines then DM performs better than RM
- RM always fails if DM fails

s $J_1 \ldots J_k$



Flow Graph for hyper-period

ame x

- Denote all jobs in hyperperiod of F frames as $J_1 \cdots J_F$
- Vertices:
  - k job vertices $J_1, J_2, \cdots, J_k$
  - F frame vertices $x, y, \cdots, z$
- Edges:
  - (source, J_i) with capacity $C_i = e_i$
    - Encode jobs' compute requirements
  - (J_i, x) with capacity f iff J_i can be scheduled i
    - encode periods and deadlines
    - edge connected job node and frame node i
      1. job arrives **before** or at the starting time
      2. job's absolute deadline **larger** or equal t
  - (f, sink) with capacity f
    - encodes limited computational capacity in

Priority Inheritance Protocol (PIP)

idea: increase the priorites only upon resource contention

avoid NPCS drawback

would still run into deadlock (think of RR task resource access)

idea: find k such that ...

Priority Inversion

critical sections to avoid race condition

Higher priority task can be blocked by a lower priority task due to resource contention

shows how resource contention can delay completion of higher priority tasks

- access shared resources guarded by Mutex or semaphores
- access non-preemptive subsystems (storage, networks)

Resource Access Control

mutex

serially reusable: a resource cannot be interrupted

If a job wants to use k_i units of resources R_i, it executes a lock $L(R_i; k_i)$, and unlocks $U(R_i; k_i)$ once it finished

Non-preemptive Critical Section Protocol (NPCS)

idea: schedule all critical sections non-preemptively

general solution for RM-schedulability

The time demand function for task i; $1 \leq i \leq n$:

$$\omega_i(t) = \sum_{k=1}^{i} \lceil \frac{t}{p_k} \rceil e_k \leq t$$

$$\because 0 \leq t \leq p_i$$

holds a time instant t chosen as $t = k_j p_j, (j = 1, \cdots, i)$ and $k_j = 1, \cdots, \lfloor \frac{p_i}{p_j} \rfloor$

rate-monotonic

- running on uniprocessor, tasks are preemptive, no OS overhead for preemption

task T_i has higher priority than task T_j if $p_i < p_j$

schedulability test for RM (Test 1)

Given n periodic processes, independent and preemptable, $D_i \geq p_i$ for all processe
periods of all tasks are *integer* multiples of each other

a sufficient condition for tasks to be scheduled on uniprocessor: $U = \sum_{i=1}^{n} \frac{e_i}{p_i} \leq 1$

schedulability test for RM (Test 2)

sufficient but not necessary condition is $U \leq n \cdot (2^{\frac{1}{n}} - 1)$ for n periodic tasks

or $n \to \infty$, we have $U < \ln(2) \approx 0.693$

earliest-deadline first (EDF)

depends on closeness of absolute deadlines

Priority Inheritance Protocol (PIP)

idea: increase the priorities only upon resource contention

avoid NPCS drawback

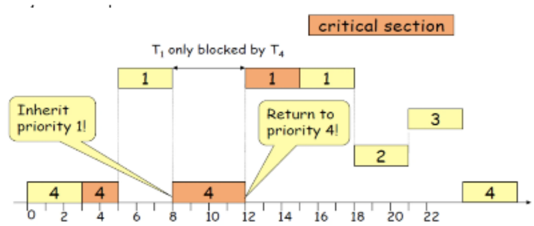
would still run into deadlock (think of RR task resource access)

xecutes at a priority higher than the priorities of all

nly when some lower priority job is in critical

of tasks

riority task for a long time even without resource



rules:

- When a task T1 is blocked due to non availability of a resource that it needs, the task T2 that holds the resource and consequently blocks T1, and T2 inherits the current priority of task T1.
- T2 executes at the inherited priority until it releases R.
- Upon the release of R, the priority o... it held when it acquired the resource

EDF schedulability test 1

set of n periodic tasks, each whose relative deadline is equal to or greater than its period
iff $\sum_{i=1}^{n} (\frac{e_i}{p_i}) \leq 1$

EDF schedulability test 2

relative deadlines are not equal to or greater than their periods

$$\sum_{i=1}^{n} (\frac{e_i}{\min(D_i, p_i)}) \leq 1$$

Priority Ceiling Protocol (PCP)

idea: extends PIP to prevent deadlocks

- If lower priority task TL blocks a higher priority task TH, priority(TL) ← priority(TH)
- When TL releases a resource, it returns to its normal priority if it doesn't block any task. Or it returns to the highest priority of the tasks waiting for a resource held by TL
- Transitive
  - T1 blocked by T2: priority(T2) ← priority(T1)
  - T2 blocked by T3: priority(T3) ← priority(T1)

iling(R): highest priority. Each resource has fixed priority ceiling

$$G(z) = \frac{U(z)}{E(z)} = \frac{(z+1)(z-0.9512)}{(z-0.9039)(z-0.8616)}$$

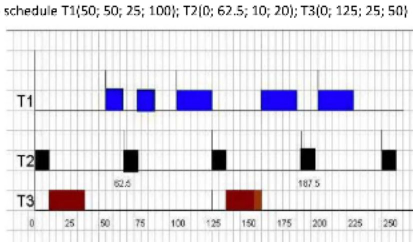
$$\therefore \frac{U(z)}{E(z)} = \frac{z^2 + 0.0488z - 0.9512}{z^2 + 1.7655z - 0.7788} = \frac{1 + 0.0488z^{-1} - 0.9512z^{-2}}{1 + 1.7655z^{-1} - 0.7788z^{-2}}$$

So $U(z)(1 + 1.7655z^{-1} - 0.7788z^{-2}) = E(z)(1 + 0.0488z^{-1} - 0.9512z^{-2})$

By applying inverse Z-Transform to the above equation, the result is:

$$u(k) = e(k) + 0.0488e(k-1) - 0.9512e(k-2)$$
$$-1.7655u(k-1) + 0.7788u(k-2)$$

Jsing DM to schedule T1(50; 50; 25; 100); T2(0; 62.5; 10; 20); T3(0; 125; 25; 50)



PCP