Informatique Embarqu e
M2 ! 201"

#rdonnancement, Tem$s-r e%

Introduction

- #rdonnancement&
  - 'hoisir $rochaine t(che qui $ourra uti%iser %e $rocesseur
    -)atch &-*
    - Tem$s-$arta+
    - Tem$s-r e%
- ,-st.mes embarqu s&
  - Mono-$rocesseur, tem$s-r e%
- /oir &
  - htt$&!!etr00.%oria.fr!s%ides!1eudi!ET200-34.$df

Attendre un 5 nement

- , o%utions
  - Po%%in+6! Attente acti5e &
    -
    -
  - A%ternati5e6&
    -
  - Interru$tion
    - 7uand %8 5 nement est $r9t en5oi d8un si+na% mat rie% au $rocesseur et e: cution du code associ .

Interru$tions

- ;5 nement mat rie% si+na%ant un chan+ement d8 tat d8un $ ri$h rique, requ rant un traitement de %a $art du s-st.me.
- E:em$%es&
  - Arri5 e d8une trame Ethernet sur un contr<%eur
    - 3ire %a trame, %a $asser = %a $i%e T'P!IP... et dire au contr<%eur que %a trame t r cu$ r e, ... $ermettre au contr<%eur de si+na%er %8arri5 e d8une nou5e%%e trame
  - >ne demande d8 criture sur un disque est termin e
    - #n n8a $%us besoin de +arder %es donn es en m moire, on $eut demander un nou5eau tra5ai% au disque ?%ecture! criture d8un b%oc*

Interruptions

- Événement déclenché par composant matériel
  - Fin d'entrée/sortie d'un périphérique
  - Échéance de temps terminée
- Signal physique envoyé au processeur de l'extérieur (contrôleur périphérique, horloge..)
  - Par l'intermédiaire d'un PIC
    - Programmable Interrupt Controller
  - Partageable ou non
- Événement asynchrone
  - Indépendant du programme courant
  - Masquable par le système (prise en compte différée jusqu'au démasquage)

Traitement Exceptions/Interruptions (1)

- Processeur suspend l'exécution en cours
- Sauvegarde état courant
  - PC, SP, registre d'état
  - Dans des registres dédiés ou dans une pile
- Sur certains processeurs (ARM), ou utilise des « shadow registers »
  - 2 jeux de registres, un par mode d'exécution

Traitement Exceptions/Interruptions (2)

- Charge nouveau contexte d'exécution :
  - Registre d'état avec mode « superviseur »
  - PC avec adresse mémoire spécifique
- Opérations de sauvegarde et de chargement
  - Non-interruptible
    - Sinon état non cohérent
    - Sauf par NMI, sur PowerPC par exemple
  - Si exception durant opérations (« double-faute »)
    - => arrêt CPU ou traitement spécial (Intel)

Exceptions/Interruptions "Vector Table"

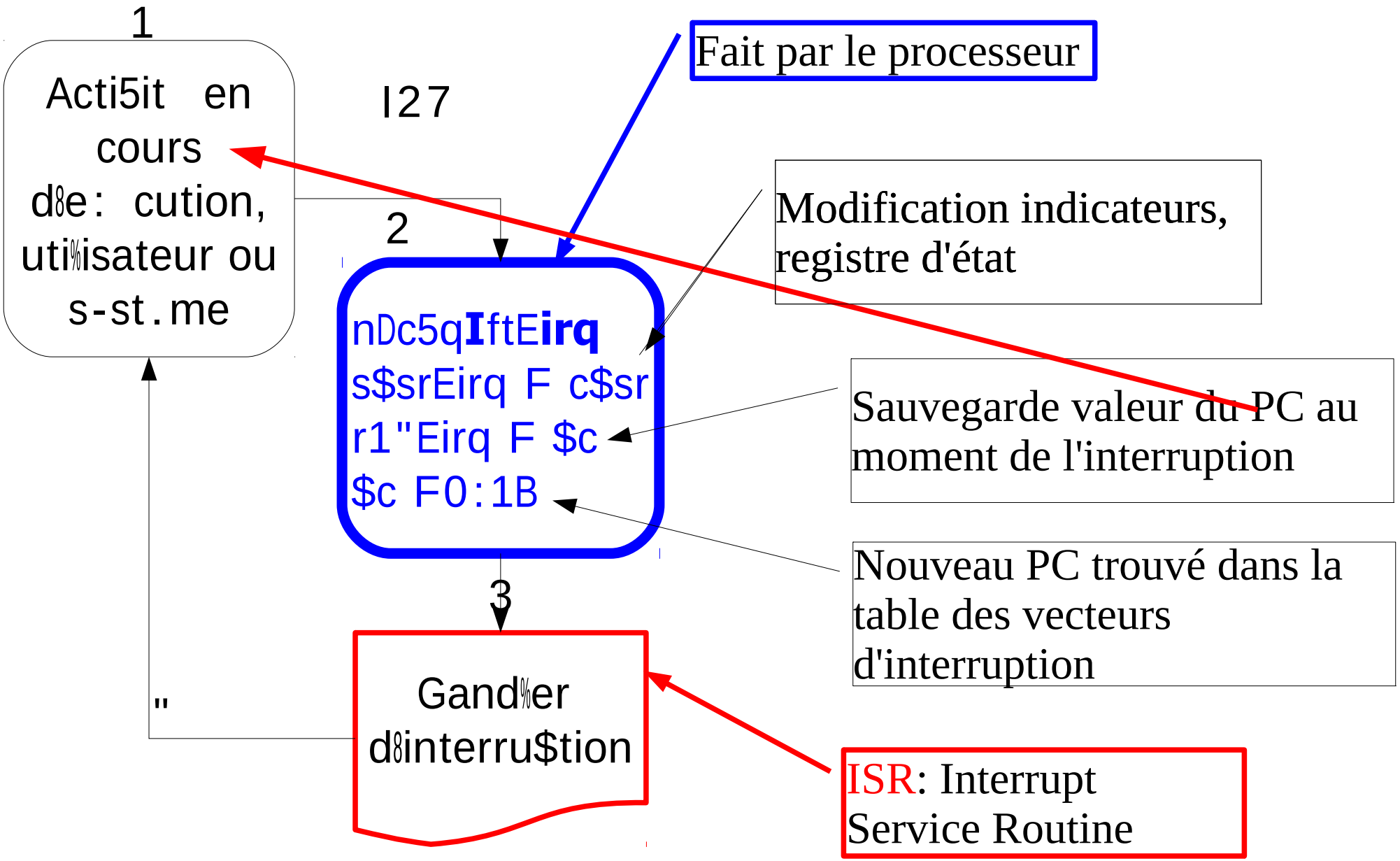
- Adresse de branchement
  - Dépend de l'exception/interruption
- Vers une table de vecteurs localisée
  - ARM: à adresse prédéfinie (0x00000000 ou 0xFFFFFFFF)
  - INTEL: dans un registre (IDTR)
- Contient une instruction de branchement
- Table de vecteurs initialisée par le système

/ector Tab%e ?A2M*

- 3a tab%e de 5ecteur contient une instruction et non $as une adresse,
- En + n ra%& instruction de branchement, sur une fonction a$$ro$ri e.

2eset	0@00000000
>ndefined Instruction	0@0000000"
,oftAare Interru$t	0@0000000B
Prefetch Abort	0@0000000'
Cata Abort	0@00000010
2eser5ed	0@0000001"
Interru$t request	0@0000001B
Fast Interru$t 2equest	0@0000001'

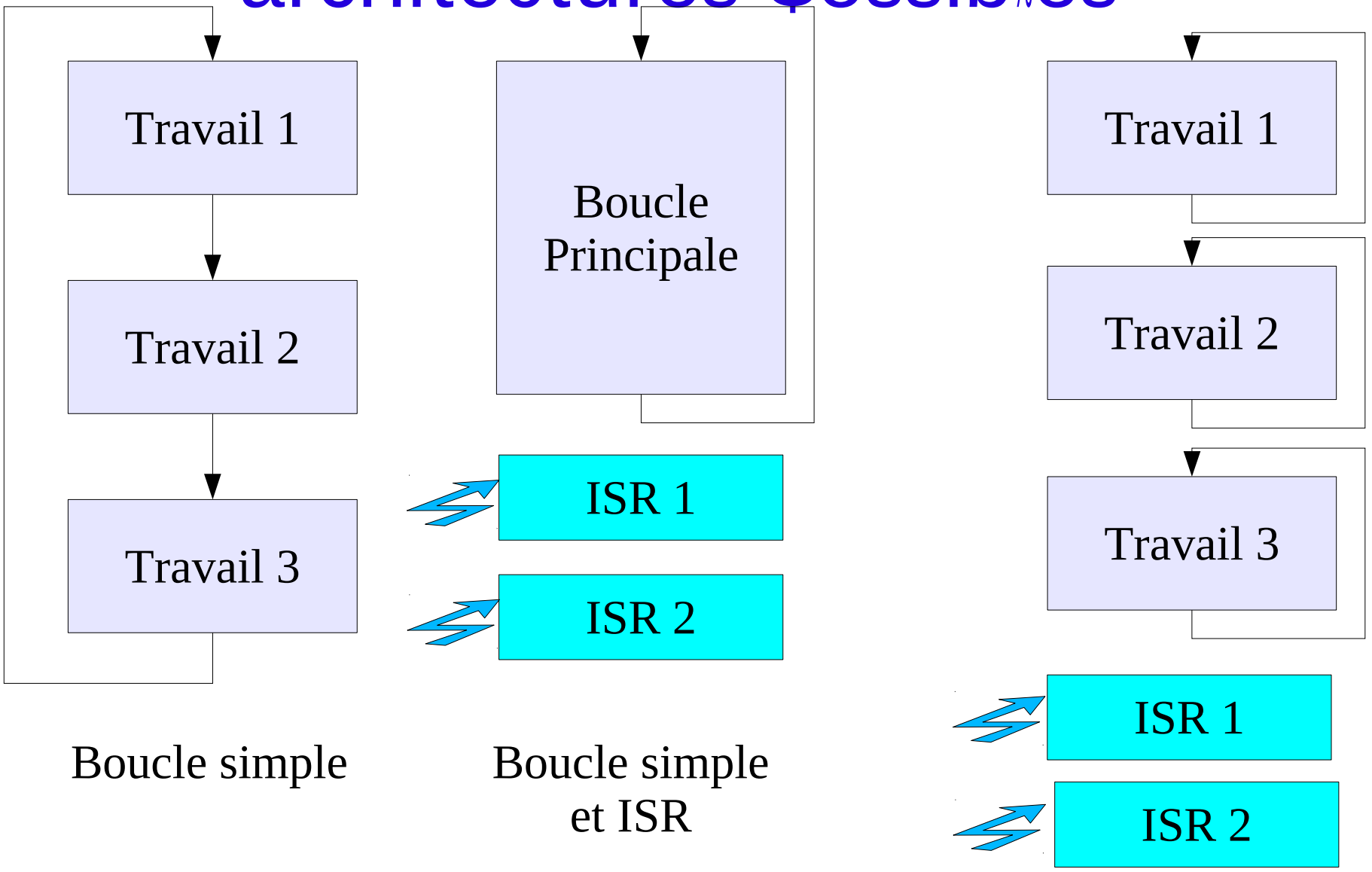
Interru$tion ! A2M



,i+nau:, en mode uti%isateur

```
#./sig
Dring!
Dring!
Dring!
Dring!
Dring!
```

A$$%ication Tem$s-r e% architectures $ossib%es RTOS

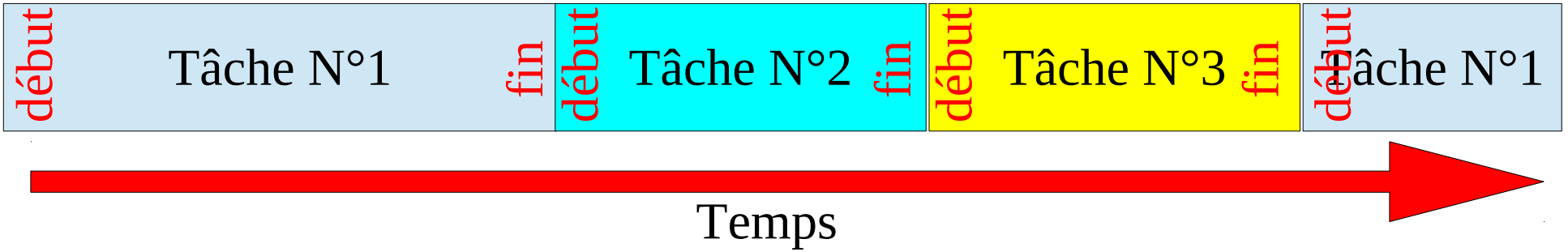


Cf http://www.embedded.com/design/programming-languages-and-tools/4434261/A-multitasking-kernel-in-as-close-as-you-can-get-to-one-line-of-code

#rdonnancement

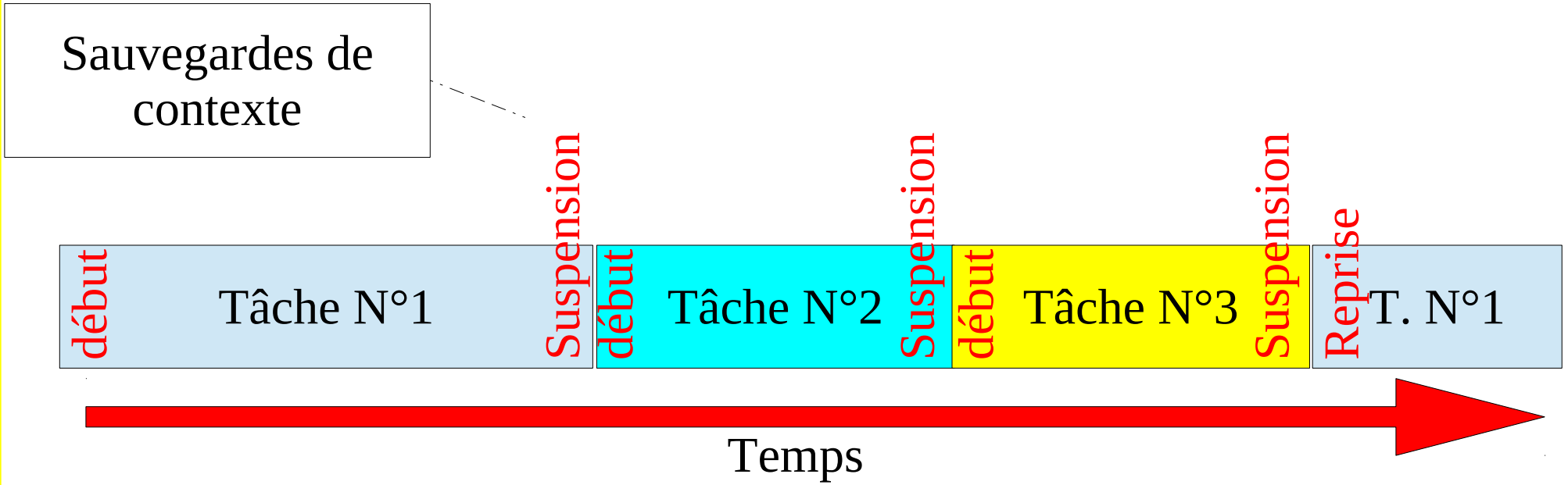
- I% - a beaucou$ de $ossibi%it s dans %es $o%itiques $ossib%es et dans %eurs im$% mentations.
- I% n8est $as forc ment n cessaire d8a5oir un # , $our $roc der = un ordonnancement
  - >ne %ibrairie $eut suffire
  - #n $eut m9me en5isa+er que ce soit fait au ni5eau mat rie%.
  - #n $eut a5oir un # ,

2un to com$%etion



- #n $ermet au: t(ches de s8e: cuter 1usqu8= %eur terminaison
  - ,imi%aire = une +rande bouc%e encha-nant %es t(ches %es unes a$r.s %es autres
  - Cu cou$, on r initia%ise %es t(ches = chaque in5ocation6&-?

A tour de r<%e



- >ne t(che s8e: cute 1usqu8= ce qu8 l
  - E%%e fasse un a$$e% b%oquant,
  - E%%e re%(che 5o%ontairement %e $rocesseur ?-ie%d*, etc
- E%%e re$rend son e: cution %= oJ e%%e %8a5ait %aiss e

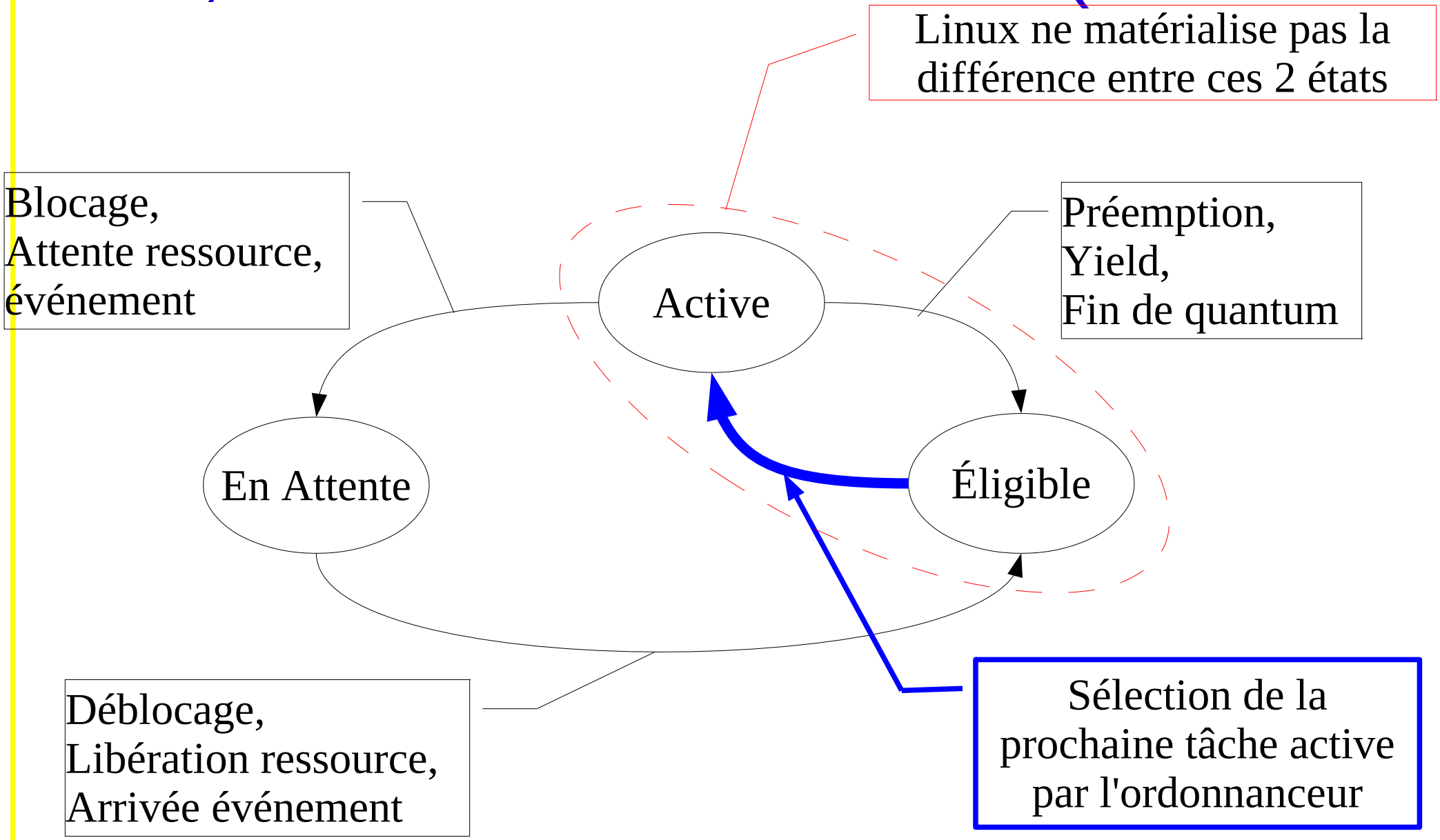
,au5e+arde, restauration de conte:tes

- Ce mani.re na5e6&
  - %8ensemb%e des re+istres ?des 5a%eurs contenues dans* au moment oJ on 5eut sus$endre %8e: cution
    - ,ommet de $i%e, com$teur ordina%, etc, I
    - E:6& TasL Cescri$tor $rocesseurs :BM
  - 3e reste de %8 tat est contenu dans %a m moire et donc $r ser5 .
- Cans %es faits, on $eut a%% +er %a sau5e+arde
  - 2e+istres ne contenant rien d8uti%e, d 1= sau5e+ard s
  - #u ne sau5e+arder qu8au moment oJ on aura besoin du re+istre...

,au5e+arde, restauration mode uti%isateur6N

- /oir aussi , et 'ie.

;tats ?sim$%ifi s* d8une t(che



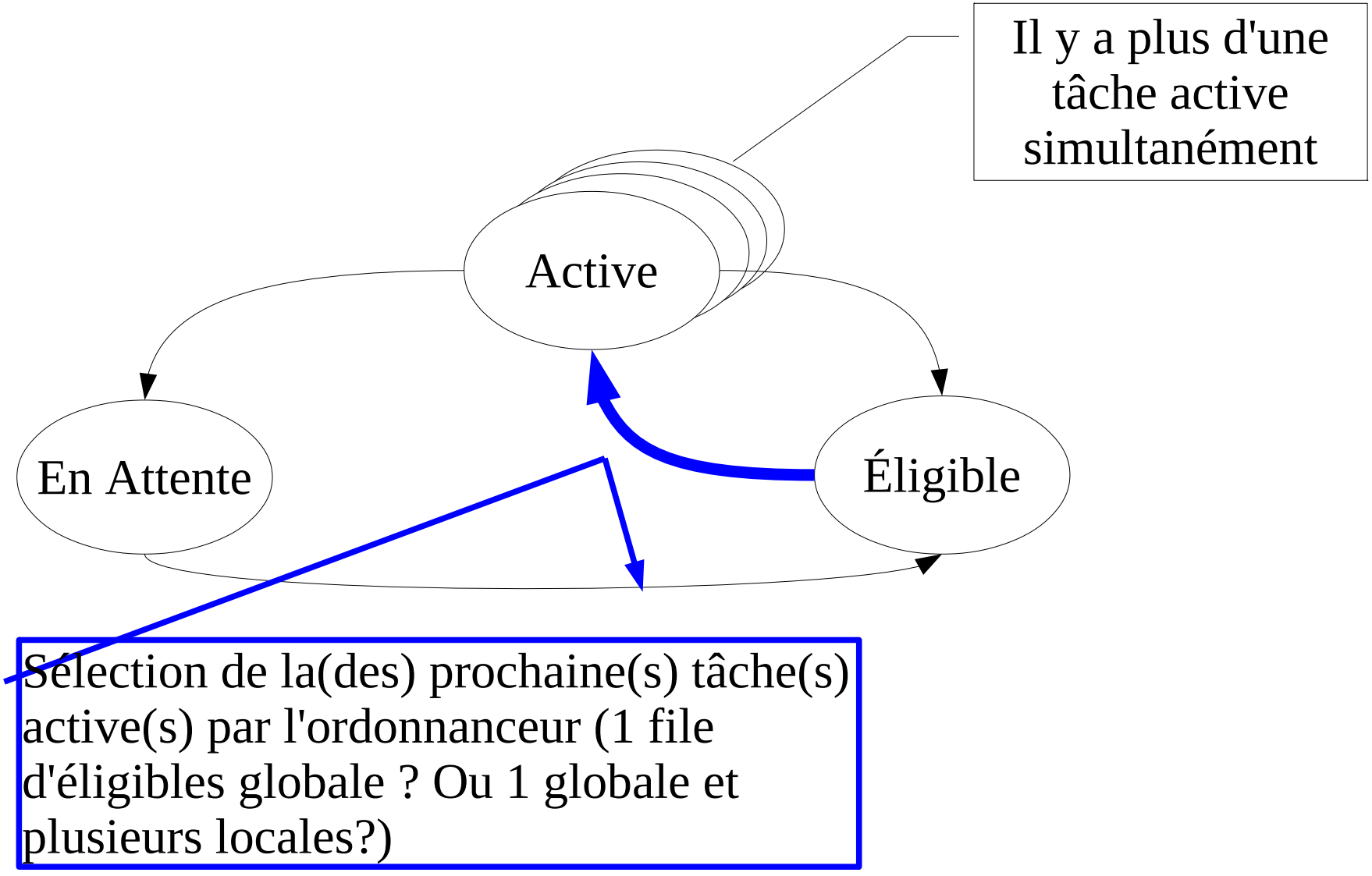
#rdonnancement, dimensions

- Po%itique ?;ta$e QR1*&
  - 7ue% est %a $rochaine t(che = obtenir %e $rocesseur6S
  - 7ue%s crit.res $our %a choisir6S
- M canique6?;ta$e QR2* &
  - ,au5e+arder %e conte:te de %a t(che qui $erd %e $rocesseur
  - 2estaurer %e conte:te de %a t(che qui obtient %e $rocesseur
- Tem$ore%%e6&
  - 7uand %8ordonnancement $eut-i% 9tre fait6S

Mu%ti$rocesseurs

- P%usieurs $rocesseurs sur une m9me machine
  - M moire $arta+ e entre %es $rocesseurs
  - Tem$s d8acc.s uniforme = %a m moire ou non ?Q>MA*
- P%usieurs $uces ou $%usieurs c T urs sur une m9me $uce, ou $%usieurs ?h-$er* U6threads6V mat rie%%es sur un m9me c T ur

;tats ?sim$%ifi s* d8une t(che



U6Attacher6V une t(che = un $rocesseur

- #n $ar%e d8affinit ?'P> affinit-*
- 'ommande6&
  -
- A$$e% s-st.me
  -
  -

Pr em$tion

- Arr9ter, sus$endre momentan ment %8e: cution d8une t(che au $rofit d8une autre. E:em$%es&
  - 2 $ondre au t % $hone a%ors que %8on est en train de faire coursN
  - 2 $ondre = une question au mi%ieu d8une e:$%ication
- #n re$rendra %8e: cution de %a t(che U6$r em$t e6V $%us tard.
- Pour qu8une $r em$tion ait %ieu, i% faut un 5 nement e:t rieur&
  - Fin d8entr e!sortie, interru$tion hor%o+e,...

Pr em$tion

- I% - a des moments oJ %e fait d8 9tre $r em$t $ourrait conduire = des incoh rences
  - Mise = 1our ! 3ecture de 5ariab%es +%oba%es acc d es $ar %a t(che ?%e code* e: cut $endant %a $r em$tion.
  - 'as usue%& mise = 1our d8une %iste $ar une t(che $r em$t e, et uti%isation de cette %iste $ar %e code e: cut au cours de %a $r em$tion.
- I% faut donc $ou5oir se $rot +er contre %a $r em$tion si n cessaire.
- Introduction d8une %atence ?d8un d %ai* dans %a $rise en com$te de %W 5 nement

#rdonnancement Tem$s-2 e%

- T(ches connues = %8a5ance ?s-st.me ferm *
  - U6dimensionnement6V du s-st.me
  - , quen0a+e manue%
- A$$roche U6$ire cas6V ?Aorst case*
  - Pas bas sur une mo-enne
  - Pas bas sur tests ou estimation
  - Identifier %es cas $ires et confi+urer %e s-st.me $our qu8i% fonctionne dans ces cas
    - I% fonctionnera donc dans n8im$orte que% autre cas

, -nchrone ! as-nchrone

- Ceu: t-$es de s-st.mes&
  - , -nchrone&
    - E:istence d8une base de tem$s commune,
    - 3es 5 nements n8arri5ent $as n8im$orte quand,
  - As-nchrone&
    - Pas d8h-$oth.se sur %es instants oJ %es 5 nements $eu5ent se $roduire

' £ ë ¯ 4 ó ¶ ¡7 2 & +0•" +AÀ + 1w `p

C terminisme

- Pou5oir +arantir que %e s-st.me res$ectera ses s$ cifications, notamment tem$ore%%es, $endant sa dur e de 5ie
  - 2 -e: cution donne des r su%tats identiques
- M thodo%o+ie
  - ,$ cifications FX $rob%.mes = r soudre
  - C terminer %es cas $ires
  - 'onditions de faisabi%it ?'F*
  - C terminer 5a%eurs num riques 'F
  - / rifier

'aract ristiques

- Mod.%es&
  - de t(ches
  - de contraintes tem$ore%%es
  - d8ordonnancement

T(ches

- T(ches concr.tes&
  - on im$ose un sc nario dWacti5ation $articu%ier des t(ches
- T(ches non concr.tes&
  - on ne connaHt $as = $riori %es instants dWacti5ation ?ou de $remi.re acti5ation* des t(ches.

T(ches

- Cur e ?$ire cas* d8une t(che τ_i& '$_i$
  - T(che seu%e sans interru$tion #,
  - Par ana%-se ou $ar mesure
- ?Pire* tem$s de r $onse d8une t(che τ_i& 2$_i$
  - Tem$s entre demande acti5ation et r $onse
  - Prend en com$te& d %ai dans e: cution induit $ar %es autres t(ches
- 2$_i$ Y '$_i$

T(ches

- 4i+ue
  - C %ai entre arri5 e d8un 5 nement et sa $rise en com$te
  - 4i+ue de τ_i& 4_i

T(ches

- P riodiques
  - Cemande acti5ation tous %es T_i ?%a $ riode*
- ,$oradiques
  - Cemande acti5ation 5ariab%e Y T_i
- A$ riodiques
  - 1 acti5ation $endant %a 5ie du s-st.me
- Fen9tr es
  - Au $%us n_i sur une dur e U6+%issante6V Z_i

T(ches

- C $endances entre t(ches
  - T(ches ind $endantes
  - T(che sim$%e& code s quentie%
  - Arbre
  - '%ient ! ,er5eur
  - 4ra$he

'ontraintes Tem$ore%%es

- ;ch ance de terminaison au $%us tard
  - T(che τ_i acti5 e = instant t_i doit 9tre termin e au $%us tard = instant t_i [C_i
  - C_i & ch ance re%ati5e
  - t_i [C_i & ch ance abso%ue
- ;ch ance de d marra+e au $%us tard

' ontraintes Tem$ore%%es

- C terminer %es ' onditions de Faisabi%it
- Pire tem$s de r $onse&
  - ∀ i F 1,.. n 2$_i$ \ C$_i$
    - Tem$s de r $onse \ ch ance
  - ' har+e du $rocesseur > F]$_{iF1,n}$?' $_i$!T$_i$* \ 1
  - En $r em$tif
    - > \ n?2$^{1!n}$ ^ 1*
    - Cifficie = uti%iser dans %a r a%it
- I% e:iste d8autres a$$roches

#rdonnanceurs

- Time Cri5en&
  - #rdonnanceur d termine quand i% est in5oqu
    - En + n ra%& in5ocation $ riodique fr quence& T_{ticL}
- E5ent Cri5en& ?ce%ui que %8on 5a consid rer*
  - #rdonnanceur in5oqu sur r ce$tion d8un 5 nement

#rdonnanceurs

- #rdonnancement $ar $riorit s&
- Fi:es ?FP*
  - 3a $riorit d8une t(che est identique $our toutes ses acti5ations ?2ate Monotonic, Cead%ine Monotonic*
- C-namiques ?CP*
  - 3a $riorit d8une t(che est ca%cu% e $our chacune de ses acti5ations ?Ear%iest Cead%ine First*
- Mi:tes

#rdonnanceurs

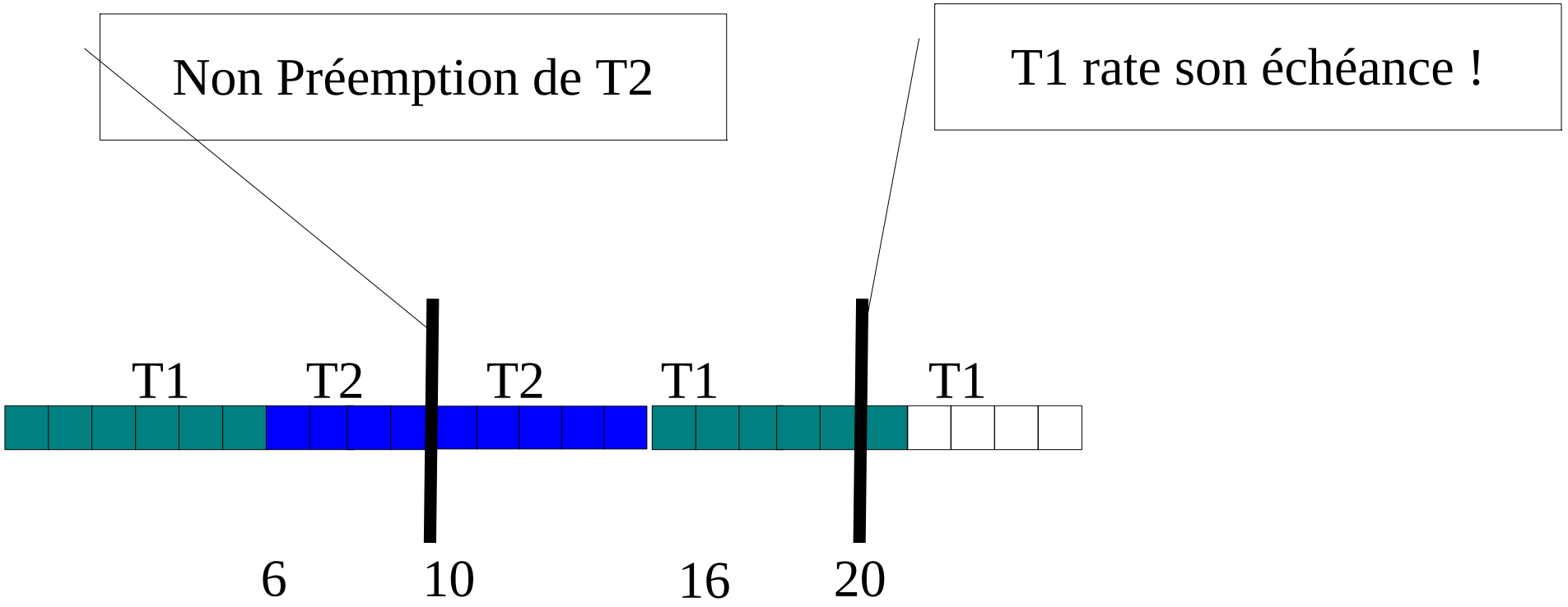
- >n a%+orithme dWordonnancement est o$tima% si i% trou5e tou1ours une so%ution $our res$ecter %es contraintes tem$ore%%es des t(ches %orsqu8i% en e:iste une.
- ,i un a%+orithme o$tima% ne trou5e $as de so%ution a%ors i% n8en e:iste $as.
- #n $eut trou5er une so%ution d8ordonnancement faisab%e a5ec un a%+orithme CP a%ors qu8i% n8e:iste $as de so%ution a5ec FP.

2ate Monotonic

- Priorit d termin e en fonction de %a $ riode
- P%us %a $ riode est $etite, $%us %a $riorit est %e5 e
- #$tima%
  - $our t(ches $ riodiques ou s$oradiques
  - A5ec ordonnancement $r em$tif

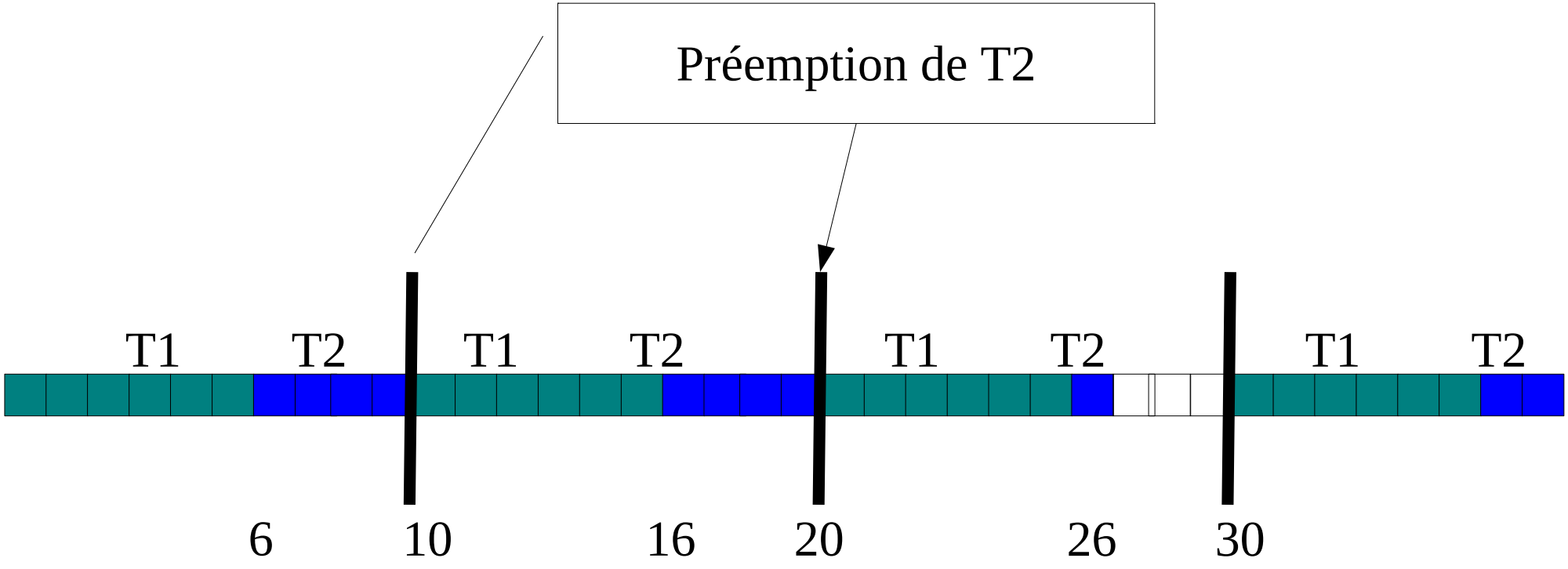
E:em$%e Qon-Pr em$tif

- T1 '1FM, P1F10, Prio1F0 ?%a $%us $rioritaire*
- T2 '2F_, P2F30, Prio2F1 ?moins $rioritaire*



E:em$%e Pr em$tif

- T1 '1FM, P1F10, Prio1F0 ?%a $%us $rioritaire*
- T2 '2F_, P2F30, Prio2F1 ?moins $rioritaire*



2M E:em$%e

- ,u$$osons 3 t(ches
  - ,ur un s-st.me oJ %a $riorit %a $%us %e5 e est 0
  - Afficha+e d8un indicateur toute %es 100 ms
    - P riode 100, 'har+e 20, Priorit 0
  - 3ecture d8un ca$teur toutes %es 200 ms
    - P riode 200, 'har+e 00, Priorit 1
  - ,ur5ei%%ance toutes %es 000 ms
    - P riode 000, 'har+e 100, Priorit 2

2M E:em$%e

- > F 20!100 [00!200 [100!000 F 0.P
- 3imite F 3?2$^{1!3}$ ` 1* F 0.PP_
- > \ 3imite
  - #a6N

FIF#

- *Pour un ni5eau de $riorit donn , %es t(ches trait es dans %eur ordre de demande d8acti5ation*
- Minimise %e tem$s de r $onse de %8ensemb%e des t(ches ?o5erhead de %8ordonnanceur minimum*
- Tem$s de r $onse ma:imum unique commun
- Pas o$tima% $our des t(ches a-ant des ch ances diff rentes

2ound 2obin ?Tourniquet*

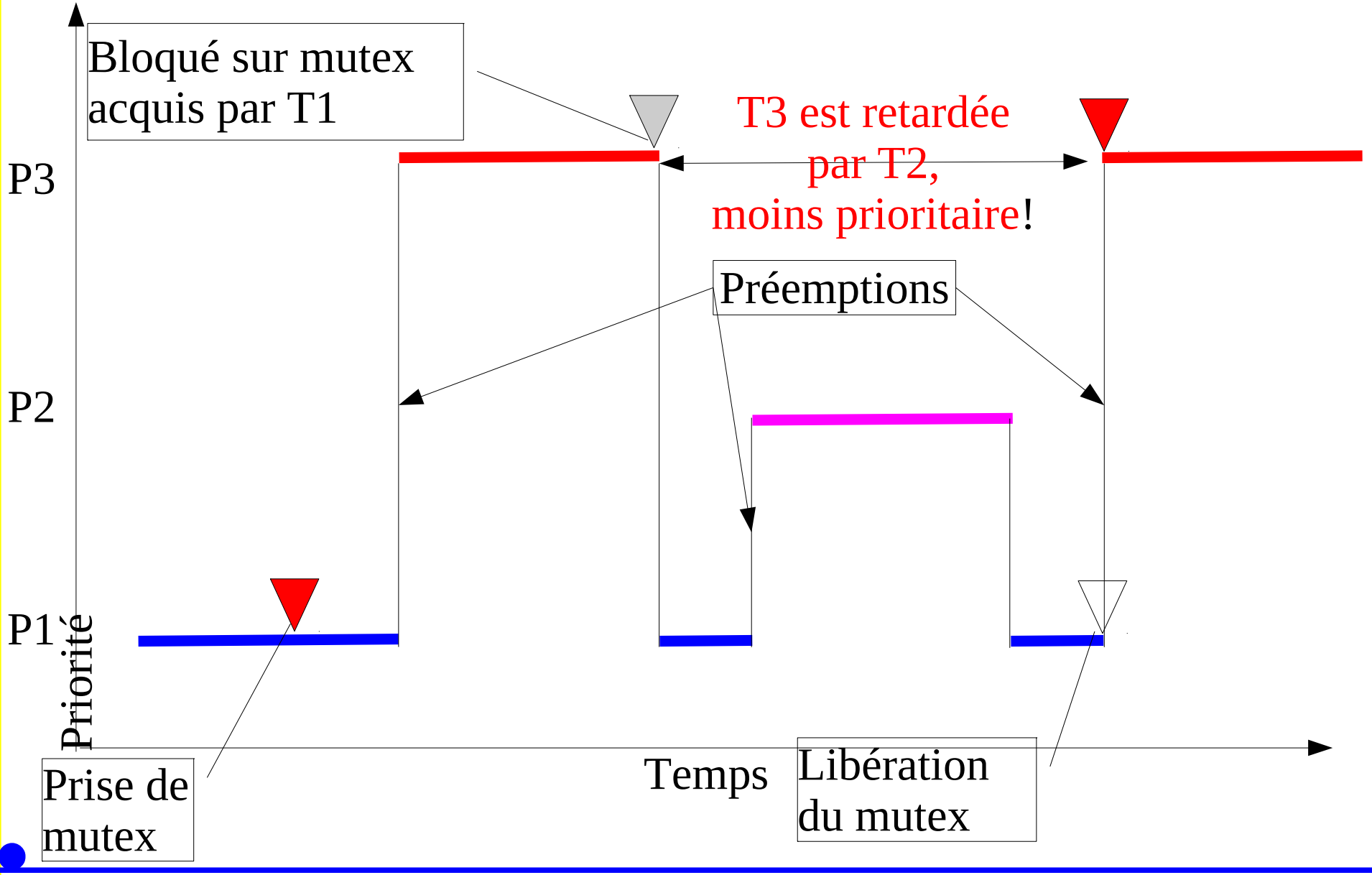
- *Pour un ni5eau de $riorit donn* , %e $rocesseur est attribu = tour de r<%e au: t(ches acti5es
- ;quitab%e
- Pas o$tima% $our t(ches tem$s-r e%
  - Mais 22 $eut trou5er une so%ution faisab%e sur certains $rob%.mes im$ossib%es en FP

Ear%iest Cead%ine First

- Priorit t(che τ_i acti5 e = t_i & t_i [C_i
- #$tima% en $r em$tif et en non $r em$tif non concret
- Tr.s $erformant
- Instab%e en cas de surchar+e& ?effet a5a%anche*
  -)ascu%er 5ers autre ordonnancement,
  - ;%iminer des t(ches
  - C +rader %e ser5ice

3inu : et ECF

In5ersion de Priorit



In5ersion de Priorit

- Pour 5iter ce +enre de situation, chan+er momentan ment %a $riorit de %a t(che T1
  - Au moment oJ e%%e acquiert %e mute:, %ui affecter une $riorit associ e = %a ressource $rot + e. T1 retrou5era sa $riorit usue%%e = %a %ib ration du mute:. U6Priorit- cei%in+ $rotoco%6V
  - Au moment oJ T3 se b%oque sur %e mute: 5erroui%% $ar T1, on fait h riter T1 de %a $riorit de T3, 1usqu8= %a %ib ration du mute: $ar T1... U6Priorti- inheritance $rotoco%6V. 'om$%e:ifie %8ordonnanceur. Transiti5it N

3inu: et %e tem$s-r e%

- 3inu: su$$orte 3 t-$es de $rocessus ! threads
  - , 'GECEQ#2MA3 & 'on5entionne%s ?tem$s-$arta+ s*
  - Tem$s-r e%&
    - , 'GECEFIF#& Priorit fi:e ?First-In First-#ut*
    - , 'GECE22& Priorit fi:e a5ec quantum de tem$s = $riorit +a%e ?2ound-2obin*
- 3es threads FIF# et 22 sont $%us $rioritaires que %es autres, et $arta+ent %a m9me $%a+e de $riorit s ?0..__*. $as affect es $ar nice.
- 3es threads tem$s $arta+ ont une $riorit 100...13_, affect e $ar nice. 3e no-au uti%ise en fait une $riorit d-namique ?uti%isation cu, tems d8attente..*

3inu: et %e tem$s-r e%

- Pour U6cr er6V un $rocessus tem$s-r e%, i% faut 9tre su$er-uti%isateur.
  - PourquoiS
- >ti%isation des a$$e%s P# , I@ $thread
  - $threadEattribute au moment de %a cr ation de %a thread
  - 'han+ement d-namique a$r.s %a cr ation de %a thread.

A$$e%s P# , I@

struct schedE$aram $aram F b.schedE$riorit- F 02 cd

$threadEattrEt attrd

$threadEattrEinit?eattr*d

$threadEattrEsetinheritsched?eattr, PTG2EACEE@P3I'ITE, 'GEC*d

$threadEattrEsetsched$o%ic-?eattr, , 'GECEFIF#*d

$threadEattrEsetsched$aram?eattr, e$aram*d

$threadEcreate?eth, eattr, efn, Q>33*d

3inu: et %e Tem$s-r e%

- 3inu: est-i% tem$s -r e%S
  - 'om$rendre&
  - 3inu: est-i% ada$t $our $ermettre = des a$$%ications a-ant des contraintes tem$s-r e% de s8e: cuter sans $rob%.me.
- 'aract ristiques attendues
  - C terminisme
  - Faib%e %atence, U6$r em$atbi%it 6V
  - #rdonnanceur a$$ro$ri
  - Ciff rents ser5ices ?timers, s-nchronisation, communication...*

3inu: et %e Tem$s-r e%

- ,ources d8informations
  -)ui%din+ Embedded 3inu: ,-stems
    - a. fa+hmour, g. Masters, 4.)en-fossef, P. 4erum
  - htt$&!!e%inu:.or+!2ea%ETime
  - htt$&!!rt.AiLi.Lerne%.or+!
  - htt$&!!free-e%ectrons.com!doc!embeddedE%inu:Erea%time.$df

3inu:, Pr em$tion

- Possibi%it de $ermettre = une t(che $rioritaire de d marrer imm diatement son e: cution
- A %8o$$os , im$oser d8attendre que %8e: cution en cours attei+ne un $oint oJ %8on $eut $roc der = in ordonnancement
- 3inu: 2."&
  - Peu de $oints de $r em$tion... E: cution en mode s-st.me ressemb%e = une U6+rosse section critique6V, ordonnancement %ors du retour 5ers %e mode uti%isateur

Prob%.mes = r soudre

- Points d8ordonnancement internes au s-st.me
- 'oht de %8ordonnancement
  - Fi:e ou roortionne% au nombre de $rocessus!threads %i+ib%es
- Traitement des interru$tions

3inu: 2.M

- Incor$oration de certaines des modifications contenue $ar %e $atch P2EEMPT-2T
  - #rdonnanceur #?1*
  - Am %ioration des $oints d8ordonnancement internes
  - Gi+h-Precision Timers
- 2este entre autres&
  - Traitement des interru$tions

Am %iorer %a Pr em$tion

- Minimiser&
  - 3a dur e des sections critiques ?$endant %esque%%es %es interru$tion sont masqu es $ar %e code de base*
    - 2 duit %a %atence de $rise en com$te
  - 3e traitement effectu sous interru$tion
    - Pendant que %8on traite une interru$tion, soit toutes %es interru$tions sont masqu es, soit %8interru$tion que %8on traite est masqu e.
    - E:& si %e traitement d8un interru$tion hor%o+e $rend $%us d8un U6ticL6V, %e s-st.me ne sera $as = %8heure et inuti%isab%e.
- Permettre un ordonnancement en retour d8interru$tion

3inu: interru$tions

- ,oftI27&
  - Faire %e minimum dans %8I,2
  - Faire %e tra5ai% r e% et U6%on+6V a$r.s a5oir acquitt %8interru$tion, a5ant de re$rendre %e tra5ai% interrom$u
  - Possib%ement faire faire ce tra5ai% $ar une thread d di e , haute $riorit
- TasL%et
  - #n $eut a5oir $%usieurs softI27 simu%tan ment ?1!'P>*
  - #n a une seu%e tasL%et ?$our un I27 donn e*, 'P> 5ariab%e

3inu:, P2EEMPT-2T

- Permettre = des threads d finies $ar %8uti%isateur de s8e: cuter = des $riorit s $%us %e5 es que %e traitement des interru$tions.
- 'r e un thread U6no-au6V $ar ni5eau d8interru$tion.
- 38administrateur s-st.me $eut a1uster %es $riorit s.
- Au+mente %a char+e du s-st.me& $assa+e ob%i+atoire $ar des threads, coht d8ordonnancement.

3inu:, P2EEMPT-2T

- AttentionN >n marteau est un tr.s bon outi%, = condition de sa5oir s8en ser5ir et de ne $as se ta$er sur %es doi+tsN
- 3e no-au 2." n8est $as $r em$tif,
  - U6Pas $ossib%e6V de tourner des a$$%ications T2
  - Modifications a$$ro$ri es maintenues ai%%eurs
- 'ertaines des modifications ont t incor$or es au no-au 2.M
  - 2este tou1ours %a n cessit d8un $atch sui5ant %es besoins a$$%icatifs

3inu: 2.M

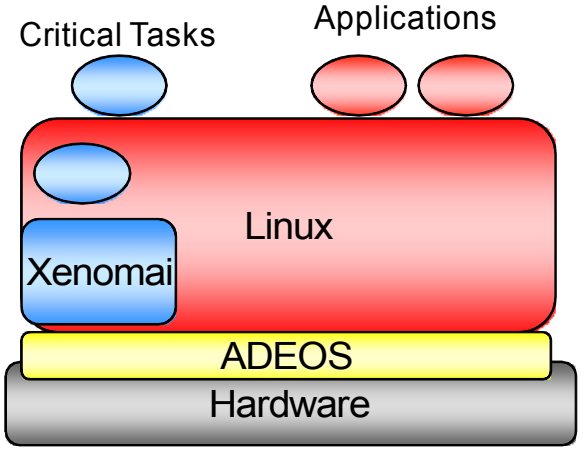
- #rdonnanceur en #?1*
  - 2em$%ac $ar 'F, de$uis 2.M.23
- #ffre 3 modes de $r em$tion ?confi+*
  - C faut ?$as de $r em$tion*
  - Pr em$tion /o%ontaire
  - Pr em$tion U6in5o%ontaire6V
    - Encore insuffisant ?- com$ris $our audio*
- 2.M.1B & h rita+e de $riorit
- 2.M.21& Gi+h-reso%ution timers ?Posi:* i 1 j sec.

Patch P2EEMPT-2T

- Introduit un ni5eau de $reem$tion U6com$%.te6V
- Permet des %atences encore $%us courtes
  i 100 j sec
- Im$act +%oba% sur $erformances
  - #n $asse $%us tem$s en chan+ement de conte:te
- Prob%.mes&
  - Pas dans %8arbre r f rence ?r sistanceN*
  - Pi%otes de $ ri$h riques = ada$ter ?quid si binaireS*

Adeos!@enomai

- ,o%ution au $rob%.me du tem$s-r e% dur&
  - #ffrir un en5ironnement 2T U6= c<t 6V de 3inu:
- E:em$%es
  - 2T3inu: ?F,M3abs, Zin2i5er*
  - 2TAI
  - Adeos !@enomai
  - ,o%utions de 5irtua%isation



Adeos !@enomai

- ACE# , ! @EQ#MAI
  - Modu%e no-au 3inu: char+ a$r.s d marra+e du no-au 3inu:
- ACE# ,
  - U6Prend %e contr<%e6V du $rocesseur
  - Interce$te e:ce$tions ! interru$tions
  - Cistribue %es interru$tions = des U6domaines6V $ar ordre de $riorit . ?5irtua%isation des interru$tions*
  - 3inu: est un domaine a5ec une $riorit faib%e

Adeos ! @enomai

- Tem$s de %atence
  - C termin $ar ACE# , %ui-m9me
  - Tem$s de U6$r em$ter6V 3inu:
    - Adeos $r em$te 3inu: k 3inu: $r em$te 3inu:
- Comaine $rioritaire & @enomai
  - E: cutif T2
    - T(ches en mode uti%isateur ! su$er5iseur
    - Ciff rentes interfaces ?sLins* dis$onib%es $our fa5oriser mi+ration de$uis des # , T2 e:istants