

ASR2-Système : les processus

Semestre 2, année 2009-2010

Département d'informatique
IUT Bordeaux-1

Avril 2010

Les processus

1. Concepts de base et définitions
 - Définitions
 - Les états des processus
 - Changements d'état
 - Multitâche préemptif / coopératif
 - Définitions (suite)
2. Les interruptions
 - Exemple : chronomètre
 - Interruptions et multitâche
3. Politiques d'ordonnancement
 - Ordonnancement
 - Critères d'évaluation
 - Tournoiquet
 - Priorités
 - Files multiples

Un processus est défini par

- un **ensemble d'instructions** à exécuter (code du progr.
- un **espace mémoire** pour les données de travail (pile, ...
- **d'autres ressources**, comme des descripteurs de fichier ouverts, des ports réseau, etc.
- des **droits d'accès**

Définitions (suite)

Définitions

- **États d'un processus** : actif, bloqué, prêt
- **contexte d'un processus** : valeur du **compteur ordinal**, **d'état**, **pointeur de pile** etc.
- **Process Control Block** : structure de données qui décrit le **contexte**
- la **table des processus** contient les PCB des processus

États des processus

Un processus peut être dans l'état

actif si il est effectivement en cours d'exécution

bloqué en attente d'un événement (fin d'E/S, délai,...)

prêt si ni bloqué, ni actif.

Actif
(en cours d'exécution)



Prêt



Bloqué
(attente d'évén.

Changements d'états

Le processus actif se bloque quand il demande une opération d'entrées-sortie.

Actif



de

à



Prêt



un processus bloqué devient prêt quand l'opération d'E/S est terminée

Actif



Prêt

entrée-sortie terminée

Définitions

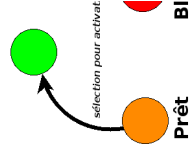
Dans un système multi-tâches un **processus** est une entité abstraite qui représente un **pc** en **cours d'exécution**

Définitions (suite)

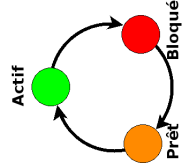
Définitions

- **États d'un processus** : actif, bloqué, prêt
- **contexte d'un processus** : valeur du **compteur ordinal**, **d'état**, **pointeur de pile** etc.
- **Process Control Block** : structure de données qui décrit le **contexte**
- la **table des processus** contient les PCB des processus

- un processus prêt peut être choisi pour devenir actif.
- il peut y avoir plusieurs processus actifs : le choix incombe à l'**ordonnanceur (scheduler)**
- diverses **politiques d'ordonnancement** sont envisageables



Transitions d'état

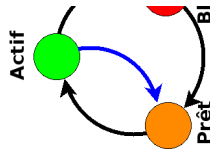


Multitâche préemptif / coopératif

Problème : cas d'un processus actif qui ne fait jamais d'E, également

Solution : le **processus actif** peut

- rendre **volontairement la main** (système **coopératif**)
- être **interrompu** au bout d'un **quantum de temps** (système **préemptif**)



Définitions (suite)

- une **commutation de contexte** se produit quand l'exéc d'un processus s'intrompt, ou reprend.
- **ordonnanceur** : composant du système, choisit un pro prêt pour l'activer.
- **quantum de temps** (dans système multitâche préemptif)
- **interruption** : signal matériel qui modifie la séquence d'exécution des instructions.

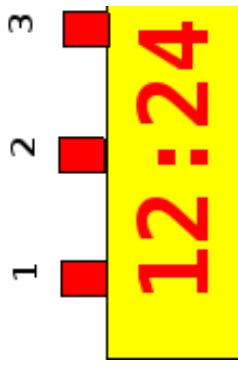
Les interruptions

- Une interruption
- est un signal qui modifie la séquence normale d'exécution des instructions par le processeur
 - déclenche
 - la **sauvegarde de l'état** du programme (quelques registres)
 - l'exécution de la **routine de traitement** de l'interrupt

Système réactif

- Les interruptions sont à la base du fonctionnement de systèmes d'exploitation
- **système réactif** qui répond aux événements causés par l'environnement (périphériques)

Exemple, chronomètre



Exemple, chronomètre

- **Programmation naïve** (sans interruption) :
 secondes = 0
 répéter indéfiniment :
 attendre 1 seconde ++
 afficher secondes
 fin → répéter
- suppose l'existence d'une fonction "attendre"
- Comment remettre le chronomètre à zéro ?

Exemple, chronometre

Utilisation d'une interruption BTN1 :

```

■ quand BTN1 se produit , lancer Remise-
secondes = 0
■ repeter indefiniment :
attendre 1 seconde
secondes ++
affichage = secondes
fin-repeter
■ routine Remise-à-zero :
centiemes = 0
secondes = 0
retour
    
```

Exemple de système réactif : chronometre

Utilisation d'une horloge (centieme de seconde)

```

■ quand BTN1 se produit , lancer Remise-
centiemes = 0
secondes = 0
■ repeter :
affichage (secondes)
fin-repeter
    
```

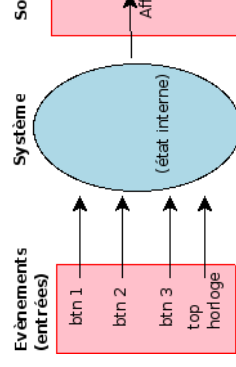
Exemple, chronometre

Utilisation d'une horloge (centieme de seconde)

```

■ routine Remise-à-zero :
secondes = 0
■ routine Top-Horloge :
centiemes ++
if (centiemes == 100) {
centiemes = 0
secondes ++
}
retour
    
```

chronometre : un système réactif



Exemple, chronometre (exercice)

Exercice :

- Ajouter un bouton STOP/RESTART
- ce bouton déclenche l'interruption BTN2
- conseil : utiliser une variable pour le "mode" (tourne,

Dans un système multitache

- Dans un système multitache, les interruptions sont causées
- des périphériques (fin d'exécution de requête)
 - des signaux d'horloge
 - des événements extérieurs
 - déroulements en cas d'erreur (accès illégal à la mémo division par zéro ...)
 - interruptions logicielles provoquées par instruction spx

Exemple : déroulement d'une interruption disque

Déroulement

- 1 mettre le processus actif à l'état prêt
- 2 déterminer la cause de l'interruption (p. ex : fin de le
- 3 trouver le processus demandeur (qui est bloqué)
- 4 lui transférer les données reçues
- 5 le remettre à l'état prêt
- 6 activer un des processus prêts

Exemple : interruption horloge

Exemple : quantum de temps épuisé (ordonnancement avec réquisition, preemptive scheduling)

Déroulement

- 1 remettre le processus actif à l'état prêt
- 2 activer un des processus prêts

L'ordonnanceur

Définition

- partie du noyau d'un système d'exploitation multitâche
- consulte la liste des processus prêts, en choisit un pour l'activer
- applique une **politique d'ordonnancement**

Quelques critères

Comment reconnaître une **bonne politique** d'ordonnancement

- équité
- efficacité
- minimiser le temps de réponse
- minimiser le temps d'exécution
- maximiser le rendement

Mais ...

Choix de politique

Mais...

- ces **objectifs** sont **contradictoire**s
 - le comportement des processus ne peut pas être prévu
 - il n'y a **pas de politique optimale**.
- On emploie donc des **heuristiques**

Tourniquet

Ordonnancement circulaire

synonymes : tourniquet, round-robin, FIFO, ...

Principe

- liste circulaire des processus prêts
- à la fin de son quantum de temps, un processus actif en fin de liste

Propriétés

- équité** garantie //
- choix du quantum : compromis entre **overhead** et // réponse

Ordonnancement avec priorités

On affecte une **priorité** (numérique) à chaque processus.

Déroulement

- choix du processus le **plus prioritaire**
- dans la même classe de **priorité** : tourniquet

Propriétés

- risque de **coalition**

Pour remédier aux problèmes d'équité, on peut faire varier les priorités (par exemple, baisse en fin de quantum).

Files multiples

Files multiples

On définit des **classes** de processus

- à chaque **classe** correspond une liste (circulaire) de pr
- chaque classe est sélectionnée régulièrement

Propriétés

- respect des **priorités**
- évite les **coalitions**