

# ASR2-Système : les processus

Semestre 2, année 2006-2007

Département d'informatique  
IUT Bordeaux 1

Février 2007

# 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
  - Ordonnanceur
  - Critères d'évaluation
  - Tourniquet
  - Priorités

# Définitions

Dans un système multi-tâches

- un **processus** est une entité abstraite qui représente un programme en cours d'exécution
- défini par
  - un ensemble d'instructions à exécuter (un programme) ;
  - un espace mémoire pour les données de travail ;
  - éventuellement, d'autres ressources, comme des descripteurs de fichiers, des ports réseau, etc.
  - des droits d'accès
- **état d'un processus** : actif, bloqué, prêt
- contexte d'un processus : valeur du compteur ordinal, pointeur de pile des registres etc.
- Process Control Block
- la table de processus contient les PCB des processus présents.

# États des processus

Un processus peut être dans l'état

**actif** si il est effectivement en cours d'exécution

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

**prêt** si ni bloqué, ni actif.

## Changements d'états

- Le processus actif se bloque quand il demande une opération d'E/S.
- un processus bloqué devient prêt quand l'opération d'E/S est terminée
- un processus prêt peut être choisi pour devenir actif.

## Multitâche préemptif / coopératif

Le processus actif cesse de l'être quand

- il se termine
- il se bloque en attente d'une opération d'entrée-sortie
- il rend volontairement la main <sup>1</sup>
- il est interrompu au bout d'un quantum de temps<sup>2</sup>

Dans les deux derniers cas, il passe à l'état **prêt**.

---

<sup>1</sup>dans un système coopératif

<sup>2</sup>dans un système préemptif

## Définitions (suite)

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

## Exemple, chronomètre

- Le code :

```
secondes = 0
```

```
répéter indéfiniment:  
    attendre 1 seconde  
    secondes ++  
    afficher secondes  
fin-répéter
```

- Comment le remettre à zéro ?



## Exemple, chronomètre

Utilisation d'une interruption BTN1 :

- quand BTN1 se produit, lancer Remise-à-zero

```
secondes = 0
```

```
répéter indéfiniment:
```

```
    attendre 1 seconde
```

```
    secondes ++
```

```
    affichage = secondes
```

```
fin-répéter
```

- routine Remise-à-zero:  
 secondes = 0

## Exemple, chronomètre

Utilisation d'une horloge (centième de seconde)

- quand CLOCK se produit, lancer Top-Horloge  
quand BTN1 se produit, lancer Remise-à-zero

```
centiemes = 0
```

```
secondes = 0
```

```
répeter:
```

```
    affichage = secondes
```

```
fin-répéter
```

## Exemple, chronomètre

Utilisation d'une horloge (centième de seconde)

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

## Exemple, chronomètre (exercice)

Exercice :

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

# Dans un système multitâche

Dans un système multitâche, interruptions

- causées par les périphériques (fin d'exécution de requête)
- causées par signaux d'horloge
- évènements extérieurs
- déroutement en cas d'erreur (accès illégal à la mémoire, division par zéro ...)
- interruptions provoquées par instruction spéciale

## Exemple : interruption disque

Exemple : interruption venant d'un contrôleur de disques

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

## Exemple : interruption horloge

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

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

# L'ordonnanceur

## Définition

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



## Quelques critères

### Critères pour 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 contradictoires
- 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

- liste circulaire des processus prêts
- à la fin de son quantum de temps, un processus actif est placé en fin de liste
- garantit l'équité
- choix du quantum : compromis entre *overhead* et temps de réponse
- synonymes : Tourniquet, round-robin, FIFO, ...

# Tourniquet

Exercice : on considère des processus A, B et C à comportements périodiques :

- calcul pendant 10 ms pour A, 10 ms pour B, 45 ms pour C
- une opération d'E/S de 20 ms

Etudiez le déroulement pendant les 100 premières ms

- pour un ordonnancement circulaire sans réquisition
- pour un ordonnancement circulaire avec réquisition (quantum 20 ms)

# Priorités

On donne une priorité (numérique) à chaque processus

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

Pour remédier à ce problème, on fait varier les priorités (par exemple, baisse en fin de quantum).