SYSTEME D'EXPLOITATION TEMPS REEL

1 Principes de bases

La conception multitâche fait coexister en mémoire plusieurs programmes, les programmes s'exécutent sous le contrôle du système d'exploitation (SE) et se partagent les différentes ressources du système. Le système d'exploitation multitâche doit alors mettre en œuvre les règles de possession et partage des ressources, définir les compétiteurs (programmes en mémoire) et les contrôler tout le long de leur exécution. L'objectif principal de la programmation multitâche est d'exploiter le plus efficacement possible toutes les ressources d'un système informatique.

1.1 Quasi parallélisme

Un système d'exploitation doit en général traiter plusieurs tâches en même temps. Comme il n'a qu'un processeur, il résout ce problème grâce à un pseudo parallélisme. Il traite une tâche à la fois, s'interrompt et passe à la suivante. La commutation des tâches étant très rapide, le microprocesseur donne l'illusion d'effectuer un traitement simultané.

Un système comportant un processeur et qui est capable d'exécuter plusieurs tâches simultanément et dit multitâche - monoprocesseur.

A l'échelle du processeur, les tâches seront exécutées alternativement, la simultanéité est donc virtuelle on parle dans ce cas de quasi-simultanéité.

1.2 Notion de Processus

1.2.1 Tâche, processus

Une tâche est une unité logicielle effectuant une action de contrôle de processus spécifique et s'exécutant sous le contrôle d'un système d'exploitation temps réel. Une Tâche en exécution est un processus. Un processus est donc l'activité résultant de l'exécution d'une tâche séquentielle avec ses données par un processeur.

1.2.2 Descripteur de Tâche

A chaque tâche est associé une structure de donnée nommée « descripteur » dans lequel le système mémorise les informations associer à cette tâche. Ces données sont nécessaires à la gestion des processus. L'ordonnanceur connait les tâches uniquement par leurs descripteurs. En effet, toutes les opérations réalisées sur une tâche sont effectués au niveau de son descripteur.

A une tâche est associé un segment de pile, pour ses besoins propres (appels des procédures, passage des paramètres..).

Dans la pratique, tous les descripteurs sont reliés entre eux selon le principe du double chaînage dans une file, de façon à intervenir rapidement sur un processus donné.

1.2.3 Etats élémentaires d'un processus

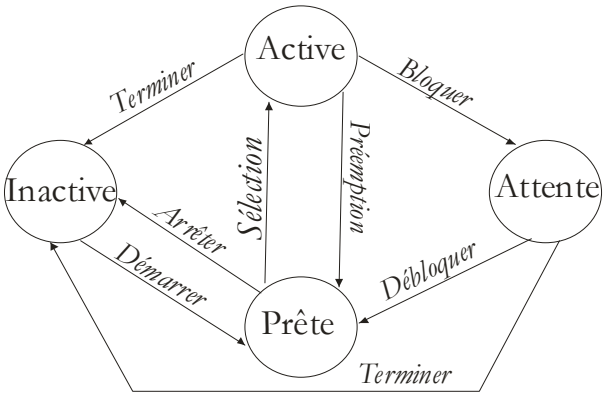
Une Tâche peut être dans l'un des états fondamentaux suivants :

- Active (Elue): la tâche est en cour d'exécution;

- Prête (ou Eligible): en possession de toute les ressources nécessaire à son fonctionnement sauf du processeur (état d'attente);

- Attente : en attente d'une ressource indispensable à son exécution futur.

- Inactive : la tâche a terminé la fonction pour laquelle elle été conçue, elle ne participe plus à l'activité du système.



- Terminer, Arrêter et Démarrer : primitives de gestion des tâches

- Bloquer, Débloquer: primitives de synchronisation.

- Préemption, Sélection : décision de l'ordonnanceur

Le processus élu est choisi parmi les processus prêts par l'ordonnanceur.

2 Coopération entre tâches

En environnement multitâche, les processus coopèrent en vue de la réalisation d'une activité commune. On distingue deux sortes de coopération, la coopération temporelle faisant intervenir les notions de blocage et de déblocage du processus ; la coopération spatiale se rapportant à l'échange d'informations entre processus. Ces deux types de coopération caractérisent respectivement la synchronisation et la communication entre processus, l'une des difficultés de la mise en œuvre du multitâche étant d'assurer le partage des ressources du système (exclusion mutuelle).

La coopération temporelle dans divers processus peut être assurée de différentes façons. On rencontre plus couramment :

- la synchronisation par sémaphore ;

- la synchronisation par événement;

2.1 Les sémaphores

Le concept de sémaphore est une solution élégante à la plupart des problèmes d'exclusion mutuelle. Ce concept nécessite la mise en œuvre d'une variable (le sémaphore), et de deux opérations atomiques associées P et V.

Un sémaphore joue le rôle d'un distributeur de ticket, que les processus utilisent à l'aide de deux opérations:

• Opération P (PRENDRE), correspond à une demande de ticket.

• Opération V (VENDRE), correspond à une libération de ticket.

A un sémaphore, on associe:

- Un compteur, qui peut prendre des valeurs entières, négatives, nulle ou positives;

- Une file d'attente de type FIFO, où sont stockés les processus en attente de ticket.

Tout processus qui prend possession de la ressource doit exécuter deux procédures:

- Une procédure P(s), au début;

- Une procédure V(s), lorsqu'il libère la ressource;

```
Procédure P(s)
Début
   S ← S – 1
   Si S ≥ 0  Alors
         Le processus prend la ressource
           Sinon
         Le processus est mis en attente
         dans la file
  Finsi
Fin
```

```
Procédure V(s)
Début
   S ← S + 1
   Si S  ≤  0  Alors
         Sélectionner un processus dans la
         file et le rendre éligible.
       Sinon
         Rien de spécial, la tâche poursuit
         son activité
  Finsi
Fin
```

Ces deux procédures sont appelées des *primitives*. Une primitive est une séquence programmée grâce à laquelle l'utilisateur peut demander au moniteur l'exécution d'une fonction déterminée.

2.2 Les événements

Une deuxième technique de synchronisation courante fait appel à la notion d'évènements lorsque ce terme désigne précisément un signal logiciel asynchrone par rapport au fonctionnement du processus récepteur. Dans ce cas l'évènement est représenté par une variable booléenne.

- Evt = 0 : évènement est dans l'état NON ARRIVE (faux).

- Evt = 1 : évènement est dans l'état ARRIVE (vrai).

On attache à l'événement une file d'attente, trois opérations essentielles sont définies : le positionnement de l'évènement (Evt = 1), sa remise à zéro (Evt = 0) et l'attente de l'évènement (insertion dans la file d'attente). Trois primitives sont associées à ces opérations : SET, RESET et WAIT.

2.3 Communication par boîtes aux lettres

Il existe plusieurs méthodes de communication entre les tâches. Les plus courantes étant celles des boîtes aux lettres ou de file de message. Ce mécanisme relève du modèle producteur-consommateur.

Une boîte aux lettres est une zone d'échange entre deux tâches. Deux files d'attente sont associées à une boîte aux lettres :

- une file d'attente des messages
- une file d'attente des tâches

Le mécanisme producteur-consommateur doit assurer une bonne régulation de la communication, c'est-à-dire éviter que le producteur dépose une donnée dans un tampon déjà plein. De même éviter que le consommateur ne vienne lire une donnée dans un tampon vide.

Les tâches utilisent les primitives :

Déposer (donnée) pour la tâche de production

Prendre (donnée) pour la tâche de production.