

Ordonnancement de Processus

Pedro Penna^{1,2}, Henrique Freitas¹,
Marcio Castro², Jean-François M  haut³

¹ Pontifical University of Minas Gerais (PUC Minas), Belo Horizonte, Brazil

² Federal University of Santa Catarina (UFSC), Florianapolis, Brazil

³ Universit   Grenoble Alpes (UGA), Polytech Grenoble, France

R  sum  

Le processus est l'abstraction cl   de tout syst  me d'exploitation. La cons  quence est qu'un syst  me d'exploitation, comme Nanvix, repose sur ce concept. Pour cette   tape, vous travaillerez sur le module principal du sous-syst  me de gestion des processus de Nanvix, l'ordonnanceur de processus (*process scheduler*). Au d  part, vous   tudierez et analyserez l'impl  mentation fournie avec Nanvix, et vous proposerez et impl  menterez ensuite des extensions    cet ordonnanceur.

Contexte

Nanvix est un syst  me d'exploitation multit  ches (*multitasking*). Pour cela, Nanvix supporte l'ex  cution concurrente de plusieurs programmes. Pour fournir cette fonctionnalit  , comme d'autres syst  mes d'exploitation, Nanvix s'appuie sur les processus, une abstraction de programme en cours d'ex  cution (*running program*), qui encapsule le flot d'ex  cution, les variables, la pile d'ex  cution et toutes les informations importantes qui concernent le programme lui-m  me. Avec cette abstraction, le travail du syst  me d'exploitation est d'attribuer le processeur    chacun des processus en cours d'ex  cution sur le syst  me. Le noyau du syst  me doit donc   tre capable de changer le contexte du processeur pour passer rapidement d'un processus    un autre. Pour effectuer ce changement de contexte entre deux processus, le syst  me va avoir besoin de restaurer les registres du processeur ainsi que l'environnement m  moire avec les donn  es et variables que le processus admis    s'ex  cuter va manipuler. De cette mani  re, le syst  me d'exploitation va donner l'illusion que les programmes s'ex  cutent simultan  ment.

Cependant, une phase importante de l'ordonnancement est la sélection du nouveau processus qui va s'exécuter sur le processeur. Avec plusieurs processus présents dans le système, il y a plusieurs choix potentiels pour activer un nouveau processus :

- Celui qui attend depuis le plus longtemps,
- Celui qui a presque terminé son exécution
- Celui qui a la plus haute priorité.

Le choix que va faire l'ordonnanceur aura une grande importance sur l'interactivité, la performance l'équité pour le partage des ressources de la plate-forme.

Description du travail

L'ordonnement de processus du système Nanvix suit une politique simple de tourniquet (*round robin*) : l'ordonnanceur affecte à chaque processus un quantum de temps et il ordonnancera l'ensemble des processus en tourniquet. Pour faire cela, le système d'exploitation conserve pour chaque processus un compteur qui indique la date depuis laquelle le processus attend d'être exécuté. Quand le quantum de temps du processus courant arrive à expiration, l'ordonnanceur choisit le processus qui attend depuis le plus longtemps. C'est donc ce processus, avec le plus grand temps d'attente, qui est sélectionné pour s'exécuter.

Cette stratégie est facile à implémenter et permet l'équité pour des systèmes à tâche unique. Cependant, il n'est pas suffisant pour des systèmes plus robustes qui supportent le multi-tâches. Par exemple, imaginez un scénario qui comprend un processus interactif avec un terminal (édition de texte) et 99 autres processus qui effectuent principalement du calcul. Si l'ordonnanceur affecte un quantum de temps de 100 ms (milli-seconde) à chaque processus, chaque processus va s'exécuter toutes les 10 secondes. Si cela ne pose pas de problème pour les processus consommateurs de processeur, cela est très ennuyeux pour le processus interactif et l'utilisateur vivra une mauvaise expérience et remettra peut-être en cause l'efficacité de son éditeur...

Pour résoudre ce problème, vous allez implémenter une politique d'ordonnement équitable qui sera basée sur les priorités. Pour ce nouvel ordonnancement, à chaque décision d'ordonnement, le processus avec la plus grande priorité sera sélectionné pour s'exécuter. Pour l'exemple précédent,

le processus interactif aura une plus grande priorité que les 99 processus qui font du calcul. Pour commencer, vous mettrez en place un dispositif de priorité statique. La priorité d'un processus est fixée une fois pour toute et ne change pas. Un dispositif de priorité dynamique sera proposé dans un second temps.

Vous fournirez pour cette étape un rapport décrivant votre solution et les modifications que vous avez apportées au code de Nanvix.

Nous vous fournirons des programmes d'évaluation que vous pourrez tester. Vous expliquerez les avantages et les inconvénients de votre solution.

Conseils

L'implémentation du sous-système de gestion de processus se trouve dans le répertoire `kernel/pm`. Il se décompose en plusieurs fichiers :

- `pm.c` initialise le sous-système de gestion de processus.
- `sched.c` contient l'ordonnanceur de processus.
- `sleep.c` contient l'implémentation du `sleep ()` et `wakeup ()`.

Pour cette étape, vous travaillerez principalement sur le fichier `sched.c` et plus précisément sur la routine `yield ()`. C'est le corps de cette fonction qui effectue l'ordonancement, c'est à dire qui choisit le prochain processus à exécuter.