

TP1
Parallélisme et signaux sous Unix

Présentation

On désire synchroniser un ensemble de processus parallèles à l'aide de la manipulation des signaux sous UNIX en utilisant le langage C.

On considère le calcul des nombres premiers compris entre 2 et n . La solution classique consiste à retirer les multiples de 2, retirer les multiples du nombre immédiatement supérieur à 2 et non retirés précédemment, c'est à dire 3, etc...

La solution parallèle consiste à initialiser successivement plusieurs processus parallèles: un pour les multiples de 2, un pour les multiples de 3, un pour les multiples de 5 etc... Chaque processus reçoit un premier nombre p , de son prédécesseur (nombre qui est premier, car il n'est pas multiple des nombres premiers découverts précédemment), imprime ce nombre, puis filtre les nombres suivants en éliminant les multiples de p . Les nombres non divisibles par p sont émis vers le processus suivant. Mathématiquement, les nombres plus grands que racine de n et non éliminés par les processus inférieurs à racine de n sont premiers. Il suffit donc que le nombre diviseur ne dépasse pas racine de n .

Modélisation

On utilisera uniquement 2 processus parallèles. Pour cela, on utilise un processus parent P qui enregistre les nombres dans un fichier f puis crée 2 processus $P1$ et $P2$ coopérant pour calculer les nombres premiers, puis imprime le contenu du fichier f et se termine. Le processus P coordonne les autres processus comme suit :

- Il crée $P1$ et $P2$ pour calculer à partir de f parallèlement et respectivement les nombres divisibles par 2 et par 3 (3 est le premier nombre premier déduit par $P1$, il est communiqué à $P2$ le plus tôt avant de réaliser le reste de son travail) qu'ils enregistrent respectivement dans $f1$ et $f2$.
- P attend la fin de ce travail avant de mettre à jour f en éliminant tous les nombres de $f1$ et $f2$.
- Puis le travail est refait en invoquant $P1$ et $P2$ pour calculer à partir de f parallèlement et respectivement les nombres divisibles par le premier nombre de f et par le premier nombre premier déduit par $P1$, (qui est communiqué à $P2$ le plus tôt avant de réaliser le reste de son travail) qu'ils enregistrent respectivement dans $f1$ et $f2$.
- P attend la fin de ce travail avant de mettre à jour f en éliminant tous les nombres de $f1$ et $f2$.

Ce travail est refait jusqu'à atteindre racine de n processus ($P1, P2, P1, P2, \dots$)

Travail demandé

Il s'agit de synchroniser tous ces processus à l'aide signaux pour réaliser la fonction décrite ci-dessus. Le système doit comprendre une interface (textuelle seulement, sous forme d'impressions) permettant de suivre et de comprendre le fonctionnement des différentes entités du service objet de ce TP.

Un rapport décrivant toutes les étapes de la réalisation du TP doit être rédigé.

Connaissances requises sous Unix

- ✓ Manipulation des fichiers, droits d'accès aux fichiers
- ✓ Création de processus
- ✓ Synchronisation par signaux entre processus.

Echéance

21 Novembre 2021.