

Devoir à rendre – 14 avril 2024

a) Une manière de déterminer les nombres premiers dans un intervalle est de découper l'espace de recherche en plusieurs sous-intervalles et d'affecter la recherche dans chaque intervalle à un processus créé par le père. Ainsi si on découpe l'intervalle $[1, N]$ en p sous-intervalles, le processus père lance p fils et le fils k recherche l'intervalle $[kN/p + 1, (k + 1)N/p]$, $k = 0, \dots, p-1$.

Cette technique permet, si on dispose d'une machine équipée de p processeurs, de paralléliser la recherche (et d'essayer de retourner le résultat p fois plus vite). Elle a cependant l'inconvénient d'affecter des espaces de recherche dont les temps d'exploration sont inégaux. Ainsi le processus 0 terminera la recherche sur $[1, N/p]$ bien avant le processus $p - 1$. Il (et donc un des processeurs de la machine) sera donc oisif jusqu'à la fin du programme. La charge de travail entre les processus (et donc les processeurs) est inégalement répartie.

Une solution pour répartir la charge de travail, consiste à utiliser un "pool" de p processus travailleurs à qui un processus maître affecte successivement des travaux de recherche de nombres premiers dans de petits intervalles (taille $T \ll N/p$).

Quand un travailleur fini sa recherche, le maître lui affecte la recherche dans un intervalle encore inexploré.

On désire mettre en place cette solution avec un programme en langage C. On pourra éventuellement demander à l'utilisateur la taille N et le nombre de processus p .

Le maître doit être le processus père; les p travailleurs des fils créés par le père. Pour dialoguer avec les fils, le père utilise un tube différent pour chaque fils. Pour dialoguer avec le père, les fils utilisent un tube commun.

Le dialogue sera défini comme le suivant :

- le père passe à un fils les bornes de l'intervalle de recherche par le tube associé au fils.
- le père passe deux entiers de valeur 0 pour indiquer au fils que ces travaux sont terminés et qu'il peut se terminer.
- le fils renvoie au père les nombres premiers trouvés et 0 pour indiquer qu'il a terminé l'exploration de l'intervalle.
- pour s'identifier auprès du père, un fils précède chaque nombre envoyé par son numéro.

A rendre:

- un readme expliquant la démarche que vous avez employée pour résoudre ce problème (sous-programme, algorithmes utilisés, ...); et comment compiler votre programme.
- un programme en langage C (avec des procédures et/ou des fonctions et commenté!) permettant de mettre en œuvre la solution à ce problème. Les interactions avec l'utilisateur (s'il y en a), les ouvertures des tubes devront être contrôlées.

b) Architecture de type client/serveur : Dans l'exemple ci-dessous, un serveur attend une requête de clients. Le corps de la requête comprend un tableau de caractères et un entier indiquant le PID du processus client ayant émis la requête. Le client place dans le tableau de caractères une chaîne que le serveur doit inverser avant de la renvoyer comme réponse au processus ayant émis la requête. Le type de la réponse est tout simplement le PID du client.

On lance dans un premier terminal le serveur et ce terminal reste bloqué, pour y lire les indications du serveur. Dans un second terminal, on lance un client.

On aurait pour le client :

```
Client operationnel
--> Bonjour
Requete de 11648 envoyee: ->Bonjour<-
11648: reponse du serveur ->ruojnoB<-
--> █
```

Et pour le serveur :

```
Serveur: requete ->Bonjour<- du processus 11648
Serveur: tmp=7
Serveur: rep.chaine ->ruojnoB<-
Serveur: reponse envoyee a 11648
█
```

Programmer l'application ci-dessus, en faisant preuve de créativité (présentation, forme des messages ,...) et en faisant en sorte que la communication se fasse en utilisant une file de messages et/ou une mémoire partagée.

Puis programmer une autre application avec un client avec un mode de communication de votre choix qui adresse des questions au serveur qui répond aux questions et affiche les réponses. Les questions sont des opérations mathématiques simples à effectuer (par exemple « combien font $7 * 13$? »). On commencera par définir la structure contenant les questions, c'est-à-dire une opération (+,-,* et /) et deux opérands.

A rendre:

- un readme expliquant la démarche que vous avez employée pour résoudre ce problème (sous-programme, algorithmes utilisés, ...); et comment compiler votre programme.
- un programme en langage C (avec des procédures et/ou des fonctions et qui est commenté) permettant de mettre en œuvre la solution à ce problème. Les moyens de communication utilisés doivent être expliqués.