

Devoir

Exercice 1 (Token ring).

Un réseau en anneau à jeton (token ring) est un réseau informatique dans lequel pour émettre sur le réseau, un ordinateur doit posséder un jeton. Le réseau a schématiquement une forme d'anneau et un unique jeton circule entre les ordinateurs, toujours dans le même sens, de proches en proches (à gauche sur la figure 1). L'ordinateur qui possède le jeton (ici le numéro 2) est le seul qui puisse émettre, les autres recevant alors ses émissions.

En réalité, la topologie physique du réseau est en étoile dont le centre est un matériel réalisant l'interconnexion et la transmission des informations ainsi que la circulation du jeton (à droite sur la figure 1). Mais pour cet exercice nous ne nous intéressons qu'à la vue schématique en anneau.

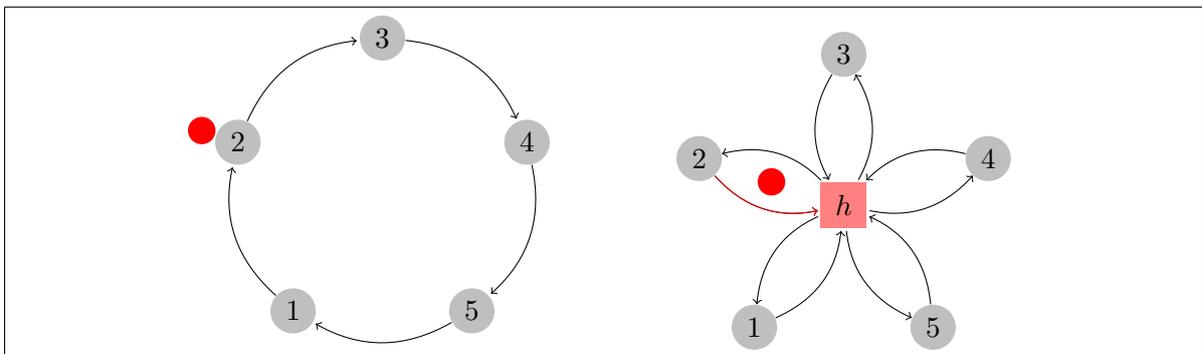


Fig. 1: Réseau en anneau à jeton, vue schématique et topologie réelle

Le but de cet exercice est de simuler schématiquement un anneau à jeton avec des processus au lieu d'ordinateurs et des tubes pour réaliser les connexions en cycle. Attention : on ne représente pas la topologie réelle en étoile mais bien la topologie schématique en cycle. Le processus qui aura le jeton sera supposé effectuer une tâche non précisée – dont on imagine qu'elle est de la plus haute importance et qu'elle nécessite un accès exclusif à une ressource – avant de faire passer le jeton au processus suivant.

Les nœuds du réseau seront créés dynamiquement en fonction d'un entier n (strictement positif) donné en paramètre à votre programme (argument à la ligne de commande).

Au départ, le premier processus branchera sa sortie standard sur son entrée standard, à l'aide d'un tube, réalisant ainsi l'anneau à un nœud. Puis le processus écrira $n - 1$ sur sa sortie standard (figure 1).

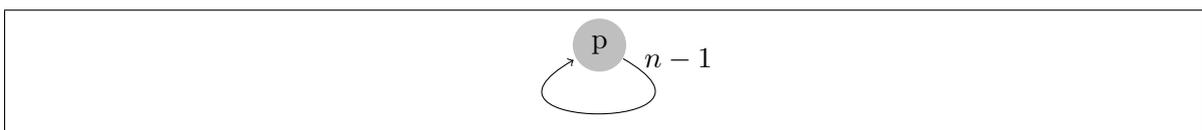


Fig. 2: Premier nœud : le processus initial juste après la première écriture

Un nouveau nœud (processus) sera créé de la façon suivante. À chaque fois qu'un nœud recevra un entier $k > 0$ sur son entrée il se dupliquera par un `fork()` en insérant son clone

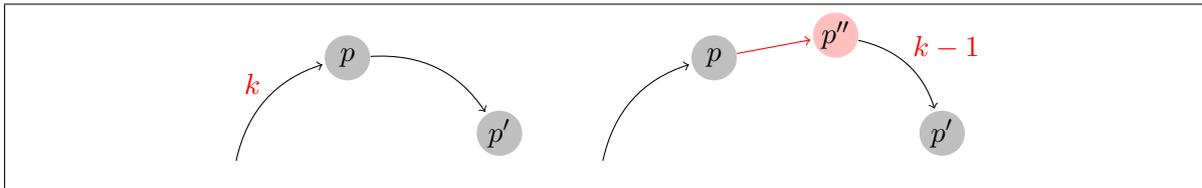


Fig. 3: Création d'un nœud (p'') par passage du jeton (avant, après)

dans l'anneau et passera l'entier $k - 1$ au processus suivant (figure 1). La connexion se fera à chaque fois entre sortie standard et entrée standard.

Une fois que n aura atteint la valeur zéro, cet entier servira de jeton pour le réseau en anneau. Chaque nœud recevant le jeton effectuera la tâche (appeler une fonction `tache()`, ne réalisant aucune instruction) et passera le jeton au nœud suivant. Et ceci indéfiniment.

Vous ferez en sorte que taper `Ctrl-C` dans votre terminal affiche le nombre de tours effectués par le jeton depuis le début (utiliser la sortie d'erreur standard). Une seconde frappe de `Ctrl-C` moins de deux secondes après la première détruira tous les processus de l'anneau.

Question : ordre de création et arbre généalogique. Pour un anneau à 8 nœuds dans quel ordre seront créés les nœuds ? Répondre en numérotant les nœuds de l'anneau par dates de création croissantes. Donner également l'arbre généalogique des huit processus.

Question : nombre de cycles. Créer un réseau en anneau à 100 nœuds. Une fois le réseau complètement créé combien de tours complets fait le jeton en 10 secondes ? En déduire une majoration du nombre de cycles d'horloge (utiliser la cadence processeurs) nécessaire à faire passer le jeton d'un processus à son voisin immédiat. Vous pouvez modifier votre code de manière à effectuer cette mesure. Bien entendu, le résultat dépend de la machine hôte et de son activité. Essayez de réduire cette dernière au minimum. Si on pouvait refaire cette mesure pour 1.000, 100.000 nœuds, devrait-on s'attendre à trouver toujours le même nombre de cycles d'horloge ? Discuter.

Quelques recommandations. Prenez bien soin de tester vos appels systèmes et en particulier de détruire tous les processus si un `fork()` venait à échouer. Écrivez de petites fonctions (une quinzaine !) plutôt que de tout faire rentrer dans le corps du `main()`. L'ensemble de ce travail ne devrait vous prendre que quelques heures.

Remise du devoir. Vous rendrez votre devoir avant le 12/0100:00 CET 2008 (le soir du vendredi de la rentrée à minuit) sous la forme d'une archive (utiliser un format ouvert : `tar`, `tgz`, `zip`, `7-zip`, ...). Vous pourrez l'envoyer par mail : `boudes@lipn.fr` ou la déposer sur une page web (disons `http://lipn.fr/~boudes/se/`). Votre archive ne devra pas excéder 300Ko.

L'archive contiendra le code source et un document court (format ouvert : `pdf`, ...) de réponse aux questions. Vous pouvez vous contenter d'un seul fichier source, sinon fournissez un `Makefile`. La lisibilité du code (pas seulement la présence de commentaires) sera fortement appréciée.