

Le problème du serveur de fichiers

Dans un réseau d'entreprise, il existe un serveur central où les employés se connectent régulièrement pour télécharger des fichiers critiques. En raison des limitations de capacité du serveur et pour éviter une possible surchauffe, il y a un nombre fixe de connexions disponibles pour les téléchargements à tout moment.

Un gestionnaire de serveur est chargé de préparer les fichiers pour chaque employé avant que ces derniers puissent les télécharger. Le gestionnaire de serveur ne peut préparer qu'un seul fichier à la fois.

Le système fonctionne ainsi :

- Le serveur a un nombre limité de connexions disponibles que les employés peuvent utiliser pour se connecter et télécharger leurs fichiers.
- Si un employé arrive et qu'il reste des connexions disponibles, il se connecte et attend que le gestionnaire prépare son fichier pour le télécharger.
- Si toutes les connexions sont occupées, l'employé doit attendre qu'une connexion se libère.
- Si le gestionnaire de serveur est inactif (comme un barbier endormi), il est réveillé lorsqu'un employé arrive.
- Le gestionnaire prépare les fichiers un par un, et lorsqu'un fichier est prêt, l'employé peut procéder à son téléchargement.

De la même manière que pour le problème du barbier nous avons émis des contraintes :

- Les threads des employés doivent invoquer une fonction nommée `getFile`.
- Si un thread employé arrive lorsque le serveur est saturé (toutes les connexions sont prises), il doit invoquer une fonction `waitForConnection` qui le bloque jusqu'à ce qu'une connexion se libère.
- Le thread du gestionnaire de serveur doit invoquer `prepareFile`.
- Lorsqu'un gestionnaire invoque `prepareFile`, il ne doit y avoir qu'un seul thread employé invoquant `getFile` en même temps.