

Le système de fichiers UNIX (suite)

Etienne Lozes

ENS Cachan

Différents types de fichiers

On en a déjà (un peu) parlé

- fichiers réguliers (fichiers disques "classique")
- répertoires
- fichiers spéciaux (périphériques, disques, partitions,...)
- liens symboliques

Reprennons les deux derniers types de fichiers

- tubes
- sockets

La communication interprocessus

Un des grands thèmes du système

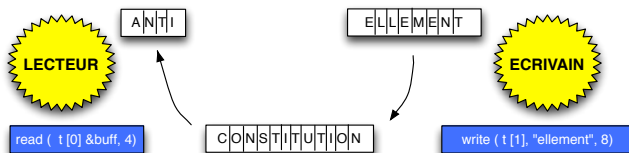
Pour quoi faire ?

- partager des données (calcul parallèle, ...)
- séquencer un traitement (pipeline)
- alerter (ex : find parallèle), voire préempter (ex : kill).
- envoyer une requête, y répondre (clients/serveurs)
- ...

De nombreuses approches

- **celle d'aujourd'hui** : des fichiers dédiés aux communications
- bien d'autres (signaux, code de retour, mécanismes de synchronisation,...)

Les tubes : communication interprocessus *asymétrique*.



Buffer à deux têtes : lecture et écriture. \approx file FIFO

- Communication en *mode connecté*.
- Taille bornée \Rightarrow écriture parfois bloquante.
- Communication en *mode flot* (à opposer au mode datagramme).

Question

Si on fait la même chose avec un fichier régulier, quelles différences observe-t-on ?

Les tubes : communication interprocessus *asymétrique*.

Deux modes d'utilisation

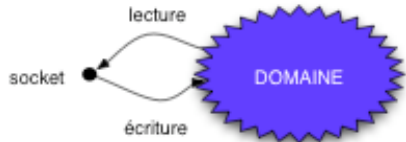
- **Tube anonyme** : ne sert qu'une fois, connu du processus qui le crée (et de ses fils).
- **Tube nommé** : plusieurs sessions (successives) possibles, connu de tous.

Début et fin de connection

- Connecté = (au moins) un lecteur et (au moins) un écrivain.
- S'il n'y a plus de lecteur, chaque écrivain est prévenu (signal SIGPIPE) quand il veut écrire.
- S'il n'y a plus d'écrivain, le tube se bouche (caractère EOF).

Communication *symétrique* : les sockets

Socket = point de communication



Contraintes immédiates

- une communication implique **deux** sockets.
- une socket doit être identifiable par une **adresse**.

Analogies

- prise de téléphone, numéro d'appel.
- compte e-mail, adresse électronique
- boîte à lettre (que le facteur peut ouvrir), adresse postale.

Notion d'adresse

Qu'est-ce qu'une adresse ?

- Une adresse identifie la socket de manière unique au sein d'un **domaine**.
- L'adresse est choisie par le processus détenteur de la socket : **attachement** d'une socket à une adresse.

⇒ pas de communication entre des sockets de domaines différents.

Exemples

- Domaine UNIX (local) : référence (comme les autres fichiers)
- Domaine Internet : une adresse IP et un port.

⇒ Démonos *sockunix* et *netstat -a -f domaine*

Divers critères de communication

Rigidité

- mode connecté : la communication est négociée et bi-partite.
- mode non connecté : adresse de destination précisée à chaque envoi.

Fluidité

- mode flot (ex : les tubes)
- mode datagramme

Garanties

- perte, duplication
- ordre

Questions

- 1 Quelles sont les dépendances entre ces critères ?
- 2 Quels sont les critères adoptés pour les analogies précédentes (téléphone, etc) ?

Communication en mode non connecté

Exemples d'applications :

- multimédias : visio-conferences, films, téléphonie IP
- transfert de descripteur (UNIX)
- souvent symétrie des agents
- souvent un traitement séquentiel de requêtes
- le plus souvent en mode datagramme.

Communication en mode non connecté



Actions socket serveur

- 1 Création et attachement

Actions socket client

Communication en mode non connecté



Actions socket serveur

- 1 Création et attachement
- 2 Réception (bloqué)

Actions socket client

Communication en mode non connecté



Actions socket serveur

- 1 Création et attachement
- 2 Réception (bloqué)

Actions socket client

- 1 Création (attachement)

Communication en mode non connecté



Actions socket serveur

- 1 Création et attachement
- 2 Réception (bloqué)

Actions socket client

- 1 Création (attachement)
- 2 Envoi

Communication en mode non connecté



Actions socket serveur

- 1 Création et attachement
- 2 Réception

Actions socket client

- 1 Création (attachement)
- 2 Envoi
- 3 Réception (bloqué)

Communication en mode non connecté



Actions socket serveur

- 1 Création et attachement
- 2 Réception
- 3 Extrait adresse émetteur

Actions socket client

- 1 Création (attachement)
- 2 Envoi
- 3 Réception (bloqué)

Communication en mode non connecté



Actions socket serveur

- 1 Création et attachement
- 2 Réception
- 3 Extrait adresse émetteur
- 4 Envoi

Actions socket client

- 1 Création (attachement)
- 2 Envoi
- 3 Réception

Communication en mode connecté

Exemples d'applications :

- sessions à distance (ssh, telnet, ...)
- serveur X
- le plus souvent communication en mode flot.
- toujours modèle client/serveur : communication négociée
- un serveur traite simultanément plusieurs clients

Communication en mode connecté



Actions socket serveur

- 1 Création et attachement

Actions socket client

Communication en mode connecté



Actions socket serveur

- 1 Création et attachement
- 2 Ecoute

Actions socket client

Communication en mode connecté



Actions socket serveur

- 1 Création et attachement
- 2 Ecoute
- 3 Acceptation (bloqué)

Actions socket client

Communication en mode connecté



Actions socket serveur

- 1 Création et attachement
- 2 Ecoute
- 3 Acceptation (bloqué)

Actions socket client

- 1 Création (attachement)

Communication en mode connecté



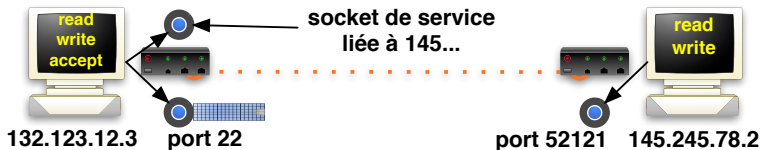
Actions socket serveur

- 1 Création et attachement
- 2 Ecoute
- 3 Acceptation (bloqué)

Actions socket client

- 1 Création (attachement)
- 2 Connect (bloqué)

Communication en mode connecté



Actions socket serveur

- 1 Création et attachement
- 2 Ecoute
- 3 Acceptation
- 4 Communication sur socket de service

Actions socket client

- 1 Création (attachement)
- 2 Connect
- 3 Communication sur socket

La mémoire partagée par projection de fichier

Projection : le fichier est chargé dans la mémoire du processus

En gros : "a = map(fichier); a[2] = a[3]" recopie le troisième caractère dans le deuxième.

Deux façons de projeter un fichier

- copie privée : les modifications ne sont pas "exportées" (mais elles sont "importées")
- zone partagée : on travaille directement sur une zone mémoire contenant "le fichier"

Applications

- zone de mémoire partagée entre processus
- bibliothèques partagées.