Examen Client/Serveur, 2000/2001
DESS CCI option Réseau

Tout document autorisé

1 Éditeur de texte

Imaginez qu'un éditeur de texte comme `Emacs` ou `VI` soit découpé en plusieurs serveurs.

- Un serveur **Document** qui reçoit des requêtes lui demandant d'envoyer ou de modifier le document qu'il contient.
- Un client **Afficheur** qui permet à l'utilisateur de faire l'édition du texte.
- Un serveur **Imprimeur** qui s'occupe de la sortie du document sur imprimante.
- Un serveur **Correcteur** qui reçoit des requêtes lui demandant si un mot est bien orthographié.

Les détails de cette architecture sont laissés volontairement flous. Vous êtes libre de faire des suppositions sur les détails à condition que vous les indiquiez.

Les questions suivantes s'appliquent à cet exemple. Il ne faut donc pas que vous donniez des réponses générales mais des réponses qui s'appuient sur cet éditeur de texte.

1. Dessinez l'architecture de l'éditeur, les noeuds sont les clients et serveurs et les arcs les connexions.
2. Quels sont les inconvénients et les avantages du découpage de l'application éditeur en clients et serveurs ? (C'est la question la plus importante du sujet.)
3. Que se passe-t-il quand 2 **Afficheur** se connectent au même serveur **Document** ?
   Proposez des améliorations pour que cela marche bien.
4. Comment faire pour transformer **Imprimeur** et **Correcteur** en clients au lieu de serveurs
5. Redessinez l'architecture.
6. Proposez d'autres clients.

2 Programmation socket

Que font ***exactement*** les serveurs suivants ? Sont-ils fonctionnels ? Sont-ils performants ? Proposez une correction du programme si nécessaire.

Le début du programme est considéré comme étant celui du TP.

—— 1 ——

```
for(;;) { s = accepte_connexion(msocket) ; write(s, "X", 2) ; close(s) ; }
```

—— 2 ——

```
for(;;) { s = accepte_connexion(msocket) ; write(s, gros_tableau, sizeof(gros_tableau)) ; close(s) ; }
```

—— 3 ——

```
for(;;) { s = accepte_connexion(msocket) ; read(s, buf, 1) ; write(s, buf, 1) ; close(s) ; }
```

—— 4 ——

```
int s[10], i, nb=0 ;
for(;;) { s[nb] = accepte_connexion(msocket) ; nb++ ; for(i=0;i<nb;i++) write(s[i], "Y", 1) ; }
```

—— 5 ——

```
s = accepte_connexion(msocket) ; t = accepte_connexion(msocket) ;
for(;;) { read(s, buf, 1) ; write(t, buf, 1) ; }
```

—— 6 ——

```
int nb ; static char t[100] = { 0 } ;
ajoute_fildes(msocket) ; ajoute_fildes(0) ;
for(;;) { s = attente_fildes() ;
          if ( s == msocket ) { t[s] = 1 ; accepte_connexion(msocket) ; }
          else { nb = read(s, buf, 100) ;
                 if ( s == 0 ) { for(i=0; i<sizeof(t);i++) if ( t[i] ) write(i, buf, nb) ; }
                 else { write(0, buf, nb) ; }
        }
```