TD 3 – Programmation avancée

# Exercice 2

Socket

Bind

(Si Service sur TCP)

Fork(2)

Newfdz accept

Select

Listen

Close tous les descripteurs sauf newfd

Setuid, Setguid

FD\_SET (service)

Exec serveur

Close(1) newfd

Fin exec

Père

Fils

* (1) Un fils à une copie de tous les descripteurs du processus parent
* (2) Un close ne s’applique qu’à un socket donné pour le processus courant

4) Parce qu’il y a deux sockets

Pas possible avec UDP car on n’

a pas de newfd, qui permet de traiter plusieurs clients à la fois

# Exercice 3

1. Gérer les messages
   1. ICMP (Internet Control Message Protocol)
      1. ICMP Echo (Ping)
      2. ICMP Reply (Ping)
   2. IGMP (Internet Group Message Protocol)
      1. Permet les communications de groupe (multicast)
2. Traiter de protocoles au-dessus d’IP inconnus du noyau
3. Écrire soit même, l’en-tête d’IP (option IP\_HDRINCL)

# Exercice 4

Ligne 1 : ouverture d’un socket UDP

Ligne 6 : positionne option TTL que l’on a précédemment définit à 1.

Ligne 12 : obtention de la date courante du système

Ligne 14 : envoie de la date à l’application serveur

Ligne 21 : Attente d’un message de la même taille que celui envoyé (écho)

Soit le serveur est sur le LAN :

Ok, soit il n’est pas atteignable 🡪 ICMP\_TIMEEXCEDEED

Soit le service n’est pas démarré 🡪 ICMP\_UNREACH

# Exercice 5

Ligne 14 : raxSocket = socket( AF\_INET, SOCKETRAW, IPPROTO\_ICMP)

Ligne 20 : n = rcvfrom(rawsocket, recvbuffer, …)

Ligne 30 : ip = (struct ip) recvbuffer;

Ligne 31 : lenIPHeader = ip->ip°hl \* 4 ; # longueur en mot de 32 bits

Ligne 33 : icmp = (struct icmp\*) (rcvbuffer + lenIPHeader)

Ligne 34 :

Ligne 35 :

Ligne 37 :