

## **M4101C : Programmation répartie**

### **TP2 : Service de communication en mode 'multicast'**

---

#### **Objectif :**

Mise en œuvre de l'**API socket** du système **Linux**.

---

#### **Travail demandé :**

Développer en langage **C**, une applications répartie gérant une **communication multicast**.

#### **Spécifications :**

##### a)- Programme **emetteur.c**

- l'adresse de groupe et le numéro de port seront passés en paramètres du main
- le nombre et la validité des paramètres seront contrôlés
- la socket locale sera configurée pour le multicast
- l'adresse de la socket distante sera configurée pour le multicast
- une suite de chaînes de caractères entrées au clavier, seront émises en multicast. Chaque chaîne sera préfixée par le nom de la machine, la date et l'heure
- le programme se terminera quand la chaîne "**stop**" sera saisie

##### b)- Programme **recepteur.c**

- l'adresse de groupe et le numéro de port seront passés en paramètres du main
- le nombre et la validité des paramètres seront contrôlés
- la socket locale sera configurée pour adhérer au groupe multicast
- un fichier texte sera créé dont le nom sera constitué du nom de la machine et du suffixe ".data"
- la suite de chaînes de caractères reçues sera stockée dans ce fichier
- le programme se terminera quand la chaîne "**stop**" sera reçue

### Exercice1 :

Sur le poste de travail, avec **gedit**, coder les 2 programmes **emetteur.c** et **recepteur.c**.  
Avec **gcc**, générer les programmes exécutables **emetteur** et **recepteur**.

Ouvrir une fenêtre terminal et lancer le programme **recepteur**.  
Ouvrir une fenêtre terminal et lancer le programme **emetteur**.  
Mettre au point les 2 programmes jusqu'à ce que l'application répartie fonctionne correctement.

### Exercice2 :

#### Note :

Demander à l'enseignant de faire créer par l'administrateur système du département, sur chaque poste de travail, une interface spéciale appelée "**tap0**" à l'@IP **192.168.1.1** qui permet à la machine hôte de communiquer avec une machine virtuelle UML sur son interface **eth0**.  
Vérifier avec la commande "**/sbin/ifconfig -a**" que cette interface est présente sur la machine hôte.

Avec la commande **uml\_lancer\_serveur**, lancer la machine **ST1** créée au TP1.  
Configurer son interface **eth0** pour qu'elle communique avec la machine hôte (cf. TP1).

Sur la machine **ST1**, sous le compte **root**, créer le répertoire **P112-TP2**.  
Sur la machine hôte, télécharger (cmde **scp**) les programmes exécutables **emetteur** et **recepteur** dans le répertoire **P112-TP2** de la machine **ST1**.

Avec la commande **uml\_lancer\_machine**, lancer la machine **ST2** créée au TP1 et créer une machine supplémentaire **ST3**.

Créer un réseau virtuel Ethernet **réseau#1** avec la commande **uml\_creer\_reseau**.  
Connecter la machine **ST1** au **réseau#1** par son interface **eth1** à l'@IP **10.0.0.1/24**.  
Connecter la machine **ST2** au **réseau#1** par son interface **eth1** à l'@IP **10.0.0.2/24**.  
Connecter la machine **ST3** au **réseau#1** par son interface **eth1** à l'@IP **10.0.0.3/24**.  
Vérifier que ces machines communiquent 2 à 2.

Consignes à respecter pour configurer l'environnement multicast des machines **ST1**, **ST2** et **ST3** :  
Exécuter la commande **ifconfig** avec l'option multicast :

```
ifconfig eth1 10.0.0.1/24 multicast
```

Définir la route permettant de donner accès aux adresses de groupe :

```
route add -net 224.0.0.0/4 dev eth1
```

Sur la machine **ST2**, sous le compte **root**, créer le répertoire **P112-TP2**.  
Sur la machine **ST3**, sous le compte **root**, créer le répertoire **P112-TP2**.

Sur la machine **ST1**, télécharger (cmde **scp**) le programme exécutable **recepteur** dans le répertoire **P112-TP2** de la machine **ST2**.  
Sur la machine **ST1**, télécharger (cmde **scp**) le programme exécutable **recepteur** dans le répertoire **P112-TP2** de la machine **ST3**.

Sur les machines **ST2** et **ST3** lancer le programme exécutable "**recepteur**".  
Sur la machine **ST1** lancer le programme exécutable "**emetteur**".

Vérifier que l'application répartie fonctionne correctement.

Arrêter les programmes sur chaque machine virtuelle.  
Arrêter les 3 machines (cmde **halt**)