Synthèse de projet transversal

Internet des Objets & NodeJS

ingesup  Mai 2015

# Membres du groupe

* REIMUND Matthias
* GLORIAN Geoffrey
* OHANA Kimberley
* RACZYNSKI Kamill

# Intitulé

Enjeux du projet:

* + Réaliser une application permettant de gérer et d’accéder à des objets sur le réseau Internet.
  + Réaliser une application web récupérant les informations de l’objet et afficher ses statistiques.

# Réaliser une application permettant de gérer et d’accéder à des objets sur le réseau Internet.

## La simulation du système embarqué sur machine virtuelle

Nous avons débuté par la mise en place d’une machine virtuelle (Vmware) sous le système d’exploitation DEBIAN en mode console.

Configuration de base :

* Mémoire : **2 Gb**
* Disque dur : **20 Gb**
* Réseau : **Bridged** les machines virtuelles peuvent communiquer totalement entre elles vers l'extérieur via la machine hôte

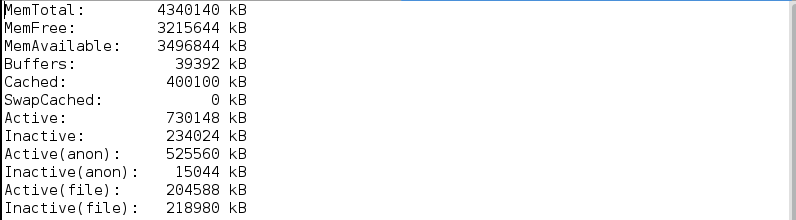
**Le serveur SSH** : Secure Shell permet à l’utilisateur d'accéder au système à distance, en rentrant leur login et leur mot de passe.

## L’Application de mesure du taux d’occupation du CPU (Central processing Unit), du taux d’occupation de la mémoire, flux entrants et sortants sur les connecteurs virtuels Ethernet.

1. Récupération des données du fichier /proc/cpuinfo/

**Concerne les informations sur le Processeur**

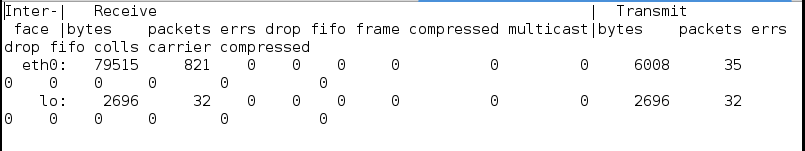
* 1. **Model\_name**
  2. **Mhz**
  3. **Adresse size**

****

1. Récupération des données du fichier /proc/net/dev/

Répertoire contenant des fichiers spéciaux nommés *devices* qui permettent le lien avec les périphériques de la machine.

Les données sur la troisième ligne eth0 représente le trafic sur l'interface eth0, en bytes entrants et en bytes sortants.



Affichage à la compilation du programme en C

* La fréquence du processeur en pourcentage.
* L’état de la mémoire en pourcentage.
* Les flux entrants et sortants sur les connecteurs virtuels Ethernet.



Remarque : la fréquence max du CPU (1.9 Ghz) or peut allez jusqu’à 2.4Ghz d’où la valeur de 126%

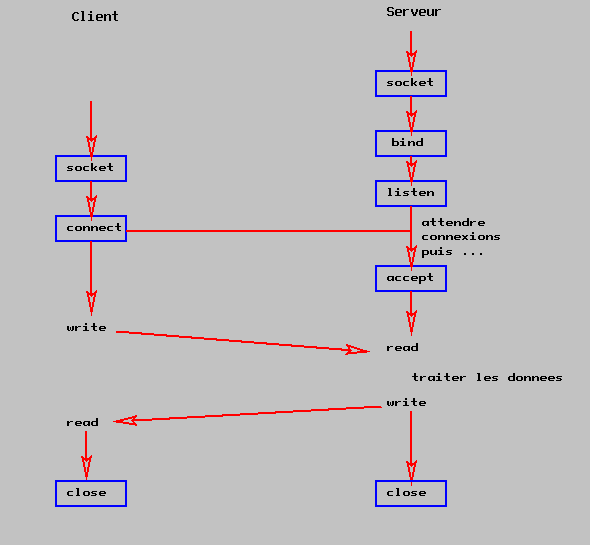
Protocole TCP/IP : gestion des accès concurrentiels

Le protocole TCP est un protocole dit connecté. Il contrôle si les paquets sont arrivés à destination. Dans le cas contraire, les paquets sont renvoyés

Les sockets sont des flux de données, permettant à des machines locales ou distantes de communiquer entre elles via des protocoles.

Un socket est décrit par:

* Un descripteur de fichier. C'est un entier qui servira aux opérations read/write ou recvfrom/sendto selon le protocole TCP.
* Des paramètres de connexion.

****

### Serveur et client : appel socket

Créer une socket c’est créer le descripteur de fichiers:

int sockfd = 0;

if((sockfd = socket(AF\_INET, SOCK\_STREAM, 0))< 0)

### Client : appel connect

Le client demande à se connecter au port P de la machine M.

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_port = htons(5001);

serv\_addr.sin\_addr.s\_addr = inet\_addr("10.137.2.9");

if(connect(sockfd, (struct s0778511572ockaddr \*)&serv\_addr,

sizeof(serv\_addr))<0)

{

L’appel ***htons*** (*Host TO Network Short*) met en format réseau l'entier court 5001.

L'appel ***inet\_addr*** génère le format ad hoc d'une adresse IP depuis une chaîne de caractères.

* + 1. **Client : read/write**

Le client se contente ensuite de faire des entrées/sortie classiques sur le descripteur de fichier,sockfd.

* + 1. **Serveur : appel bind**

Le serveur remplit une structure sockaddr\_in où il indique sur quel port il attend:

if (bind(sockfd, (struct sockaddr \*) &serv\_addr, sizeof(serv\_addr)) < 0)

{

* + 1. **Appel listen**

On indique au système que l'on attend des appels. Un paramètre supplémentaire indique la taille d'une file d'attente que peut gérer le système: quand cette file d'attente est pleine, les demandes de connexions supplémentaires sont ignorées, donc elles générent un *time-out*.

listen(sockfd,5);

clilen = sizeof(cli\_addr);

### Serveur : appel accept

Cet appel fournit une nouvelle socket, laissant donc ainsi libre la socket d'origine pour d'autres appels: c'est cette nouvelle socket qui est utilisée pour le dialogue entre le client et le serveur.

newsockfd = accept(sockfd, (struct sockaddr \*)&cli\_addr, &clilen);

# Annexe : Schéma du réseau

