

Sujet du TP n° 1

Préambule

Les supports du cours se trouvent sur Campus.

Environnement de développement

L'environnement de travail pour le cours est le suivant :

- Utilisation obligatoire du système Linux
- Édition conseillée avec **gedit**, **emacs**, etc. (ne pas utiliser **eclipse**)
- Compilation avec **gcc -Wall -o programme_executable programme.c ...**
- Outils : **make**, **gdb**, etc.

Nous vous proposons d'organiser votre répertoire de travail selon l'archive **PSE.tar**.

- Récupérez cette archive sur **Campus**.
- Ouvrez une fenêtre "Terminal", et positionnez-vous dans votre répertoire personnel : à la racine ou dans un autre répertoire où extraire l'archive.
- Extrayez l'archive avec la commande **tar xvf PSE.tar**.
- Un nouveau répertoire, **PSE**, a été créé par cette commande. Nous le nommerons **[pse]** dans les documents futurs. Il sera votre espace de travail pour les TP.

Vous trouverez dans cet espace **[pse]** :

- Un répertoire **include** contenant les fichiers **.h** des modules utilisés dans les TP. Ces fichiers contiennent la documentation des fonctions qu'ils offrent. Pour inclure tous les fichiers **.h** fournis dans les sujets des TP dans vos programmes, il vous suffira d'inclure le fichier **pse.h**.
- Un répertoire **modules** qui contient :
 - ♦ les fichiers **.c** des modules utilisés dans les TP, qui seront placés dans une bibliothèque ;
 - ♦ un fichier **Makefile** qui permet de créer la bibliothèque **libpse.a** dans le répertoire **lib**, voir le contenu de ce répertoire ci-dessous ;
 - ♦ les modules présent sont :
 - ◊ **erreur**, qui permet une gestion (simple) des affichages de messages d'erreur
 - ◊ **resolv**, qui permet d'effectuer des résolutions DNS
 - ◊ **ligne**, qui permet de lire ou d'écrire des lignes de textes à partir de fichiers
 - ◊ **msg** et **msgbox**, qui permettent de gérer des messages et les files de messages
 - ◊ **datathread**, qui permet de gérer les données privées des threads
- Un répertoire **lib** dans lequel vous trouverez (après l'avoir générée) la bibliothèque de modules **libpse.a**, utilisée dans les TP. Le fichier **Makefile.inc** présent dans **[pse]** fait le nécessaire pour lier vos applications à cette bibliothèque.
- Un répertoire **exemples** dans lequel se trouvent les exemples complets des transparents du cours. Les fichiers correspondants ont un nom sous la forme **slide999.c**, où **999** est le numéro du transparent.
- Les répertoires **TP1** à **TP8** contenant les fichiers nécessaires pour les TP. Chaque répertoire contient :
 - ♦ Un fichier **Makefile**, qui vous servira pour générer les programmes exécutables demandés dans le TP. Il utilise le fichier **Makefile.inc** décrit précédemment.
 - ♦ Selon les TP, des fichiers à compléter ou à utiliser.

Génération de la bibliothèque des modules

Placez-vous dans le répertoire **modules** et tapez **make** pour générer la bibliothèque.
Cette opération devra être refaite à chaque modification de modules de cette bibliothèque.

Arguments de la fonction main (transparent 30)

Pratique 1

- copiez le fichier **exemples/slide030.c** dans **TP1**, en lui donnant le nom **exercice1.c**.
- placez vous dans le répertoire **PSE/TP1**, tapez **make** pour générer l'exécutable et testez-le : il affiche le nombre d'arguments (il en attend au moins un, en plus du nom de la commande)

Exercice 1

modifiez le fichier **exercice1.c**, de telle sorte qu'il affiche ses arguments dans la fenêtre Terminal.

Résolution DNS (transparents 31 à 36)

Nous allons utiliser dans cette partie un module de bibliothèque placé dans le répertoire **modules**. Il réalise une résolution DNS : à partir d'un nom de domaine (par exemple **www.emse.fr**), il retourne son adresse IP. Mais avant cela, nous allons expérimenter les types et les fonctions utilisées.

Exercice 2

- copiez le fichier **exemples/slide036.c** dans **TP1**, en le nommant **exercice2.c**
- note: ajoutez l'exécutable **exercice2** dans la variable **EXE** du fichier **Makefile**.
- affichez en format hexadécimal les champs **sin_addr.s_addr** (entier 32 bits) et **sin_port** (entier 16 bits).
- rappel : l'affichage en hexadécimal d'un **int** se fait avec **"%X"** et celui d'un **short** se fait avec **"%hX"**.
- que constatez-vous ?
- corrigez en utilisant les fonctions de conversion d'un entier codé dans l'ordre du réseau en un entier codé dans l'ordre de la machine hôte, transparent 32.
- modifiez ce programme afin d'offrir les valeurs de l'hôte et du service ("www.emse.fr" et "http" en dur pour le moment) en arguments de la ligne de commande.
- par exemple : **./exercice2 www.google.com https**

Exercice 3.

- modifiez le programme précédent de manière à afficher l'adresse IP sous forme décimale à point (comme 193.49.174.194) en utilisant la fonction **inet_ntoa()**; pour cela consulter le manuel (commande man).

Pour terminer, nous allons utiliser pour la résolution DNS un module fourni dans la bibliothèque **modules**.

Exercice 4. Programmez en utilisant le module **resolv** :

- consultez le fichier **resolv.h** qui est dans le répertoire **include** pour avoir la description des fonctions, le fichier **resolv.c** étant dans le répertoire **modules**,
- dans **TP1**, écrivez un programme qui prend un hôte et un service en arguments et utilise le module **resolv** pour afficher l'adresse IP sous forme décimale à point ainsi que le port.