

# Préambule

## Introduction

Action de base du S. E.

- Gestion
- E/S périphériques
- Programmation multitâches
- Communication local et réseau  
etc...

Sous Linux, en C

8 séances. => Cours (début de la séance) puis TP (après cours) / séance.

Evaluation : Ecrite (documentation accessible, mettre sous clé USB)

Supports :

- 1 support complet
- 1 support / séances (un exercice et/ou un exemple expliquant une thématique.)

TP : Linux

Prérequis :

- Le C.
- Commande Linux basiques.

---

## Séance 1

- Installer l'archive
- Argument de la ligne de commande.
- Résolution d'adresse réseau, DNS. (Librarie compliquée)

Bibliothèque C d'appel système :

- fonction C
- action de base du S. E.
- Passage en mode noyau pour exécuter l'action, puis repasse en mode utilisateur.

## Arguments de la ligne de commande

Prenons un fichier "add.c", il permet de sommer de nombre passer en argument lors du lancement de l'exécutable.

```
/add 45 58
```

Pour faire cela, il faut modifier la fonction `main()` du programme C.

```
int main(int argc, char *argv[])
```

Le paramètre `argc` contient le nombre d'argument passé lors de l'exécution.

Le paramètre `*argv[]` est un tableau de pointeur liée à un emplacement mémoire contenant les argument passé lors de l'exécution.

/!\ Pour la commande `/add 45 58` nous obtenons 3 arguments (dans `argc`) et 3 valeur `"/add", "45", "58"` (dans `*argv[]`)

## Résolution DNS

Cela consiste à partir d'un nom de machine ou d'un nom de domaine (pour nous `"www.emse.fr"`) et d'un nom de service (dans ce cas "HTTP").

Puis de convertir ceci en une adresse IP. (pour vérifier 193.49.174.194:80 ; <adr\_ip>:)

La fonction C : `getaddrinfo()` (compliqué à utiliser)

```
getaddrinfo("www.emse.fr", "http", <a_struct>);
```

<a\_struct> -> addr ip : 32bits [193|49|174|194] : [0|1|2|3]

port : 16 bits [80]

/!\ Le codage des entiers est différents selon les systèmes,

il faut donc passer les données de la structure utilisé par des fonctions de codage pour le système.

## Codage nécessaire des entiers

- Fonction réseau : poids forts d'abord.
- PC : poids faible d'abord.

Pour faire du traitement sur les structures des fonctions réseau il faut utiliser une batterie de fonction.

Conversion NETWORK vers HOST : `ntohl()` pour 32bits, `ntohs()` pour 16bits.

Conversion HOST vers NETWORK : `htonl()` pour 32bits, `htons()` pour 16bits.