

# Plan de cours

## Architecture des Ordinateurs

### Assembleur 80386

Halim Djerroud  
révision: 0.2

Je propose que les 24h de cours **Architecture des ordinateurs** soient réparties :

- Cours : 4 séances de 2 heures = 8 heures cours
- TD : 3 séances de 2 heures = 6 séances
- TP : 3 séances : 2 séances de 2 heures + 1 séance de 6 heures = 10 heures

#### **Partie 1 : Système de numération et arithmétique binaire (6 heures)**

##### **Objectif :**

- Savoir effectuer les conversion dans les différentes bases
- Savoir comment l'information est représentée en mémoire centrale
- Savoir comment l'arithmétique est réalisée
- Savoir utiliser des opérateurs binaires en C
- Savoir identifier les types de variables et leurs tailles

##### **Cours : (2 heure)**

- Les systèmes de numération : décimal, binaire, octal et hexadécimal.
- Conversion entre bases
- Arithmétique binaire (signée et non-signée)
- représentation des flottants et la norme IEEE 754
- Représentation des caractères

##### **Contenu du TD : (2 heures)**

- Exercices à réaliser sur feuilles sur les conversions binaires

##### **Contenu du TP : (2 heures)**

- Manipuler les opérateurs binaires en langage C

## Partie 2 : La mémoire centrale (4 heures)

### Objectif :

- Connaître l'organisation de la mémoire
- Connaître l'agencement des octets et leur alignement

### Cours : (2 heures)

- Organisation de la mémoire centrale
- Caractéristiques d'une mémoire
- Contraintes d'alignement
- Agencement des octets : Endianness (little et big)
- Principe d'implémentation et câblage (facultatif)

### Contenu du TD/TP : (2h00 heures)

- Exercices à réaliser sur feuilles sur les alignements de mots mémoire et le comportement de la pile
  - Expliquer le comportement des programmes C et les choix du compilateur GCC
- 

## Partie 3 : Le CPU Intel 80386 (6 heures)

### Objectif :

- Connaître les différents registres du microprocesseur 80386
- Connaître les instructions du microprocesseur 80386
- Savoir écrire du code en assembleur
- Savoir lire la documentation
- Connaître les modes d'adressage

### Contenu du cours : (2 heures)

- Fonctionnement d'un microprocesseur
- Les registres généraux
- Le registre d'état
- Le jeu d'instructions
- Traduire en assembleur les structures conditionnelles et les structures de contrôle
- Les modes d'adressage (adressage absolu, indirect, basé et indexé)

### Contenu du TD : (4 heures)

- Exercices sur les registres du processeur
  - Écrire des programmes rudimentaires en assembleur sur feuille
  - Exercices sur les modes d'adressage
- 

## Partie 4 : Assembleur (8 heures)

### Objectif :

- Savoir coder en assembleur GNU
- Savoir mettre en place une chaîne de compilation

- Connaître les conventions d'appels de passage de paramètres
- Savoir lire la documentation
- Savoir faire des appels de fonctions de bibliothèque et des systèmes depuis l'assembleur

### **Contenu du cours : (2 heures)**

- Mise en place de la chaîne de compilation (en 32 bits)
- Organisation des fichiers assembleur (sections : text, data et bss)
- Écrire du code assembleur simple (structures conditionnelles et les structures de contrôle)
- Écrire des fonctions en assembleur
- Les conventions GCC appel de fonction assembleur depuis le langage C
- Appeler des fonctions de la *glibc*
- Utiliser les appels systèmes (en 32 bits)

### **Contenu du TP : (6 heures)**

- Mise en place de la chaîne de compilation (32 bits)
  - Écrire des programmes à l'aide de GNU AS
  - Écrire des programmes en C qui appellent des fonctions écrites en assembleur
- 

## **Prérequis**

- Savoir écrire des programmes en langage C et utiliser *glibc* (GNU C Library)
- Savoir utiliser de ligne de commande (bash)
- Savoir utiliser GCC en ligne de commande
- Savoir utiliser d'un éditeur de texte pour écrire du code source (ex : emacs, vim ou autre)
- Savoir utiliser la commande **man**, et identifier les sections
- Savoir différencier un appel système d'une fonction de la bibliothèque standard
- Savoir lire une documentation en anglais