

## TD-TP 3 : Traduction de types de données avancés

Le but de cette séance est d'apprendre à traduire les types de données suivants : pointeurs, tableaux, chaînes de caractères et structures,

Commencez par récupérer les sources pour cette séance sur Chamilo.

Les exercices sont organisés comme dans les TP précédents : un fichier `exo.c` qui contient le programme principal, un fichier `fct_exo.s` à remplir, et une règle de génération dans le Makefile (`make exo`) pour générer l'exécutable. Dans tous les exercices, il convient de vérifier l'exécution pas à pas du programme avec GDB et le simulateur.

### Ex. 1 : Manipulation de chaînes de caractères

Cet exercice vise à traduire en langage d'assemblage des programmes C manipulant des chaînes de caractères. Le fichier `fct_chaines.s` est le fichier de travail.

**Question 1** Donnez le contexte de la fonction `taille_chaine` et traduisez-la<sup>1</sup>.

**Question 2** Exécutez votre code avec le simulateur comme dans les séances précédentes. Si besoin mettez au point le programme en utilisant GDB et en affichant la chaîne résultat à la fin. Les formats d'affichage `/c` et `/s` de la commande `display` peuvent vous être utile pour afficher dans le débogueur respectivement un caractère ou une chaîne de caractères.

**Question 3** En respectant le contexte fourni, traduisez la fonction `inverse_chaine`.

### Ex. 2 : Manipulation de tableaux

Le but de cet exercice est de travailler avec des tableaux d'entiers signés 32 bits. Le fichier de travail s'appelle `fct_tableaux.s`.

**Question 1** Donnez le contexte de la fonction `tri_min` et traduisez-la.

### Ex. 3 : Manipulation de structures de données

Cet exercice permet d'appréhender la traduction de code C utilisant des types structurés (ou enregistrement).

**Question 1** Les fichiers `fct_struct.s` et `struct.c` contiennent des fonctions très simples utilisant des types structurés. Analysez ces fichiers pour répondre aux questions suivantes : Quelle convention utilise l'ABI pour passer en paramètres une structure composée de champs de 32 bits ? Comment sont représentées les structures en mémoire ? Comment accède-t-on à

---

1. On rappelle que le caractère ASCII '`\0`' vaut 0. Par ailleurs pour lire un mémoire un seul octet on utilisera l'instruction `lbu`.

ces champs dans ce cas ? Pourquoi les commentaires dans la fonction `affichage` qualifient les sauvegardes d'inutiles ?

**Question 2** Dans le fichier `fct_liste.s`, définissez le contexte de la fonction `inverse`, puis implantez-la<sup>2</sup>. Vérifiez le bon fonctionnement de votre code (`make`, `qemu`, `gdb`).

*Pour aller plus loin...* **Question 3** Implantez la fonction `decoupe` en respectant le contexte imposé, puis testez votre programme.

## *Pour aller plus loin...* **Ex. 4 : Palindrômes**

Cet exercice vous permettra de manipuler des chaînes de caractères et de faire des appels de fonctions. Le fichier de travail se nomme `fct_palin.s`.

**Question 1** Donnez le contexte de la fonction `palin`, puis implantez cette fonction. Testez enfin le résultat.

---

2. On rappelle que `NULL` est codé par la valeur 0.