Assembleur

TP 1

# Objectif :

L’objectif de ce TP est de réviser les connaissances en C et de se familiariser avec le jeu d’instruction x86 32bits.

# Prérequis :

* Connaissances en C

# Étape 1 : Révision de C

|  |
| --- |
| Quels sont les types de base du langage C ? |
|  |

|  |
| --- |
| Combien d’octets sont utilisés pour encoder les types sur une architecture 32 bits ? |
|  |
| Quelle est la taille d’un pointeur ? |
|  |

|  |
| --- |
| Quel est le bit utilisé pour stocker le signe d’un entier ? |
|  |

|  |
| --- |
| Quel est la différence entre un ‘’signed int’’ et ‘’unsigned int’’ ? |
|  |

|  |
| --- |
| Par quoi se termine une chaine de caractère en C ? |
|  |

|  |
| --- |
| Quelle est la fonction qui permet de réserver de la mémoire dynamiquement ? |
|  |

|  |
| --- |
| Complétez le tableau suivants |
| |  |  |  | | --- | --- | --- | | **Fonctions en C** | **Définition** | **Valeur de retour (le cas échéant)** | | SCANF |  |  | | FGETC |  |  | | STRCPY |  |  | | STRCAT |  |  | | STRCMP |  |  | | STRLEN |  |  | | MEMCPY |  |  | | MEMCMP |  |  | | ATOI |  |  | | RAND |  |  | | SRAND |  |  | |

|  |
| --- |
| Dessinez la représentation mémoire à l’adresse 0x42100 du tableau suivant :  char tab[20] ;  On initialise le tableau avec les éléments suivants ‘’abcdefghijklmnopqrst’’  Si chaque case mémoire représente un octet ? |
| |  |  |  | | --- | --- | --- | | Adresse | Code en C | Valeur | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
| Dessinez la représentation mémoire à l’adresse 0x42100 du même tableau mais en changeant la taille de la case mémoire à 4 octets :  Les processeur x86 utilise le little-endian[[1]](#footnote-1). |
|  |

# Étape 2 : Notions importantes

|  |
| --- |
| Quels sont les registres généraux en Assembleur x86 32 bits ? |
|  |
| Expliquez avec vos mots le fonctionnement du registre ESP : |
|  |
| Expliquez avec vos mots le fonctionnement du registres EBP : |
|  |
| Quelle est la taille du registre AL ? |
|  |

|  |
| --- |
| De quel type de liste la pile est-elle ? |
|  |

|  |
| --- |
| Dans quelle partie de la mémoire est stockée les variables locales d’une fonction ? |
|  |

|  |
| --- |
| Expliquez avec vos mots pourquoi on a besoin du tas ? |
|  |

|  |
| --- |
| Qu’est qu’une frame pour un fonction assembleur ? |
|  |

|  |
| --- |
| Quelles sont les instructions qui permettent d’ajouter ou de retirer des valeurs de la pile ? |
|  |

# Étape 2 : Pratiquer l’assembleur

|  |
| --- |
| Ecrivez en assembleur l’opération : 4 + 9 : |
|  |

|  |
| --- |
| Ecrivez en assembleur l’opération : 5 - 3 : |
|  |

|  |
| --- |
| Ecrivez en assembleur l’opération : 2 \* 10 : |
|  |

|  |
| --- |
| Quel est le résultat de l’instruction « xor eax, eax » |
|  |

|  |
| --- |
| Quelle est le résultat de l’instruction « div BL » si AX vaut 10 et BL vaut 2 ? |
|  |
| Dans quel registre se trouve le résultat ? |
|  |

|  |
| --- |
| Commet se traduit les écritures i++ et i-- en assembleur ? |
|  |

|  |
| --- |
| Comment stocker la valeur 0x1337 dans le registre EDX ? |
|  |

|  |
| --- |
| Quelle est la signification de l’instruction « MOV ESI, offset ‘’ESIEA’’ »? |
|  |
| Quelle est la représentation hexadécimal de la chaine ESIEA et esiea ? |
|  |
| Quelle est la taille en mémoire de cette chaine de caractère ? |
|  |
| Ecrivez en assembleur, l’opération pour transformer la lettre R en r ? |
|  |

|  |
| --- |
| Comment stocker le contenu du registre ESI à l’adresse mémoire 0x1000 ? |
|  |

|  |
| --- |
| Expliquez l’instruction : « MOV BYTE PTR [edi+ecx\*4], 0xAF » ? |
|  |

|  |
| --- |
| Expliquez cette instruction «MOV DWORD PTR[eax], 53494541h » ? |
|  |

|  |
| --- |
| Expliquez la différence entre MOV et LEA ? |
|  |

|  |
| --- |
| Quelle est l’instruction qui permet d’appeler des fonctions ? |
|  |
| Comment fonctionne-t-elle ? |
|  |
| Comment s’effectue le passage des arguments ? |
|  |
| En fonction de EBP, à quel offset se trouvera le 1er argument d’une fonction ? |
|  |
| En fonction de EBP, à quel offset se trouvera le 2e argument d’une fonction ? |
|  |
| En fonction de EBP, à quel offset se trouvera le 3e argument d’une fonction ? |
|  |

|  |
| --- |
| Expliquez la convention d’appel « cdecl » ? |
|  |
| Expliquez la convention d’appel « stdcall » ? |
|  |
| Dans quel registre, on passera le retour d’une fonction ? |
|  |

|  |
| --- |
| Quelles instructions représentent le if … else en C ? |
|  |
| A quoi correspond JZ ou JE ? |
|  |
| A quoi correspond JNZ ou JNE ? |
|  |
| A quoi correspond JB ou JNAE? |
|  |
| A quoi correspond JA ou JNAE ? |
|  |

1. <https://fr.wiktionary.org/wiki/little-endian> [↑](#footnote-ref-1)