

Systeme d'Exploitation

TD1 : *Get A-Way with our Horses*

Jean-Christophe Deneuille
<jean-christophe.deneuille@xlim.fr>

2 mars 2016

Exercice 1 Souviens-toi, petit scarabée!

Nous disposons d'une architecture 32 bits disposant du jeu d'instruction adapté. Pour la compréhension de ce TD, nous rappelons que les nombres utilisent la représentation du complément à 2. De plus, nous exécutons, sur l'architecture en question, le programme suivant :

```
.LC0:
8048548:                "\033[48;5;%um\n"

main:
8048424: 55                push   ebp
8048425: 89 e5             mov    ebp,esp
8048427: 83 e4 f0          and    esp,0xfffffff0
804842a: 83 ec 30          sub    esp,0x30
804842d: c7 44 24 2c 10 00 00 mov    DWORD PTR [esp+0x2c],0x10
8048434: 00
8048435: eb 01            jmp    8048438 <main+0x14> ; L3

L4:
8048437: 90                nop

L3:
8048438: b8 48 85 04 08    mov    eax,0x8048548 ; LC0
804843d: 8b 54 24 2c        mov    edx,DWORD PTR [esp+0x2c]
8048441: 89 54 24 04        mov    DWORD PTR [esp+0x4],edx
8048445: 89 04 24           mov    DWORD PTR [esp],eax
8048448: e8 e7 fe ff ff    call   8048334 ; printf
804844d: c7 04 24 50 c3 00 00 mov    DWORD PTR [esp],0xc350
8048454: e8 cb fe ff ff    call   8048324 ; usleep
8048459: 83 44 24 2c 01    add    DWORD PTR [esp+0x2c],0x1
804845e: 81 7c 24 2c e7 00 00 cmp    DWORD PTR [esp+0x2c],0xe7
8048465: 00
8048466: 76 cf            jbe    8048437 <main+0x13> ; L4
8048468: c7 44 24 2c 10 00 00 mov    DWORD PTR [esp+0x2c],0x10
804846f: 00
8048470: eb c6            jmp    8048438 <main+0x14> ; L3
```

1. Pour cette architecture, quel est la représentation décimale de chacun des entiers signés suivant ?
 - a) 0b 1111 0000
 - b) 0x CA FE
 - c) 0x 05 22
2. À quel type de jeu d'instruction avons nous à faire ici ?
3. Dérouler le programme précédent.
4. Quel sont les paramètres du `printf` (dans le programme précédent @0x8048448) ?

Exercice 2 Le bonheur est bien ici !

À présent que vous avez *bien* compris le fonctionnement du programme précédent, vous devez en écrire un (de préférence en assembleur 8086) qui *dump* une partie de la mémoire. La zone mémoire à copier débute en 0x04 09 65 67. Ces octets peuvent être copiés dans un tableau de 0xFF FF éléments qui a pour adresse 0x08 69 18 71.