TD/TP4 : Programmation Assembler x86

Nga Nguyen, EISTI

Exo 1. Boucle

Pour le code C :

```
long dw_loop(long x) {
        long y = x*x;
        long *p = &x;
        long n = 2*x;
        do {
                x +=y;
                (*p)++;
                n--;
        } while (n>0);
        return x;
}
```

Gcc génère le code assembleur suivant :

```
//x initially in %rdi
dx_loop:
        movq    %rdi, %rax
        movq    %rdi, %rcx
        imulq   %rdi, %rcx
        leaq    (%rdi, %rdi), %rdx
.L2:
        leaq    1(%rcx,%rax), %rax
        subq    $1, %rdx
        testq   %rdx, %rdx
        jg      .L2
        rep; ret
```

1. Quels registres sont utilisés pour les variables x, y et n ?
2. Comment le compilateur a traité le pointer p ?

Exo 2. Switch

Voici un code C non complet :

```
void switch_omitted (long x, long *dest) {
        long val = 0;
        switch(x) {
                // body with cases
        }
        *dest = val;
}
```

Et gcc génère le code assembleur suivant :

```
//x in %rdi
switch_omitted :
        addq    $1, %rdi
        cmpq    $8, %rdi
        ja      .L2
        jmp     *.L4, (,%rdi,8)
```

avec le jump table :

```
.L4:
        .quad   .L9
        .quad   .L5
        .quad   .L6
        .quad   .L7
        .quad   .L2
        .quad   .L7
        .quad   .L8
        .quad   .L2
        .quad   .L5
```

En utilisant ces informations, répondre aux questions suivantes :

1. Quelles sont les valeurs de cas dans le switch ?
2. Quels cas ont des multiples labels dans le code C ?

Exo 3. Transfer de données conditionnel (conditionnal move)

Considérer le code suivant :

```
long cread(long *xp) {
        return (xp ? *xp : 0);
}
```

Et une implémentation non valide qui essaie d'utiliser un transfert de données conditionnel (cmove) :

```
//invalid implementation of function
// x in %rdi
cread:
        movq    (%rdi), %rax
        testq   %rdi, %rdi
        movl    $0, %edx
        cmove   %rdx, %rax
        ret
```

Écrire une fonction C `cread_alt` qui a le même comportement que `cread` mais qui peut être compilé en utilisant cmove (transfert conditionnel de données).