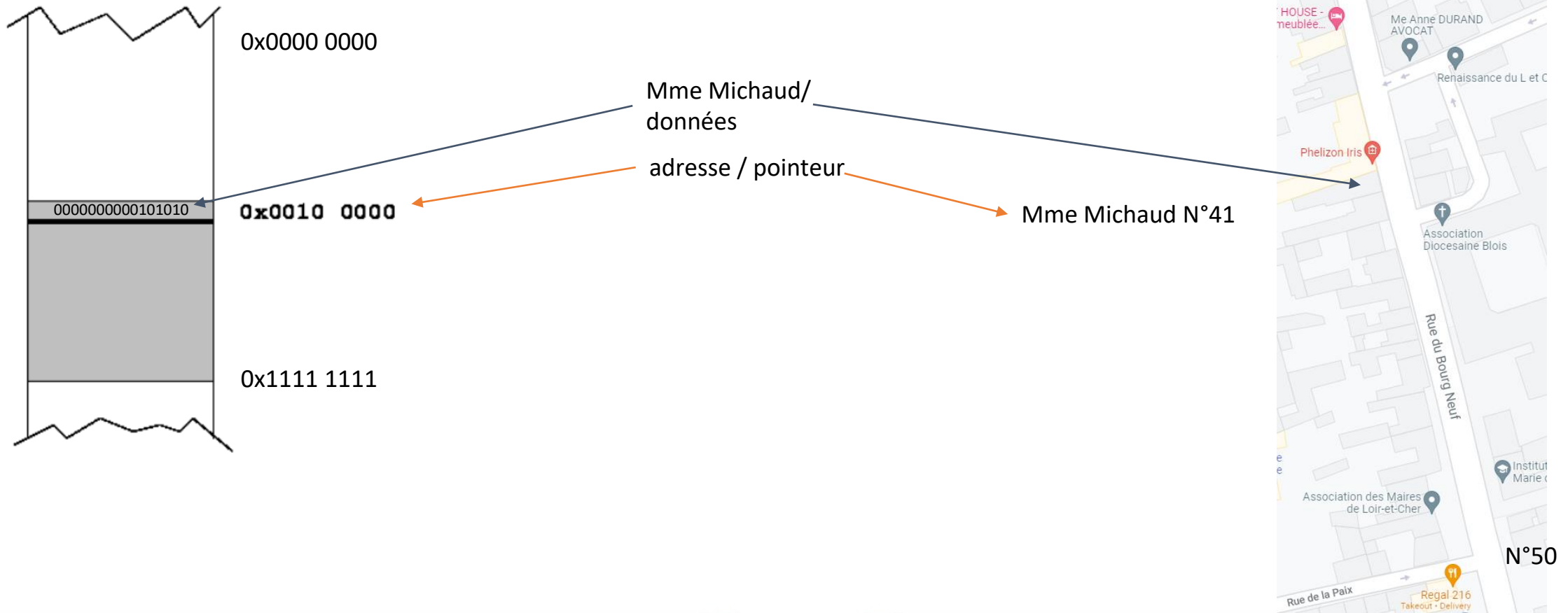


# Architecture des Ordinateurs (5)

Evelyne Moreau – Alexandre Chanson

# Mémoire

# Rue

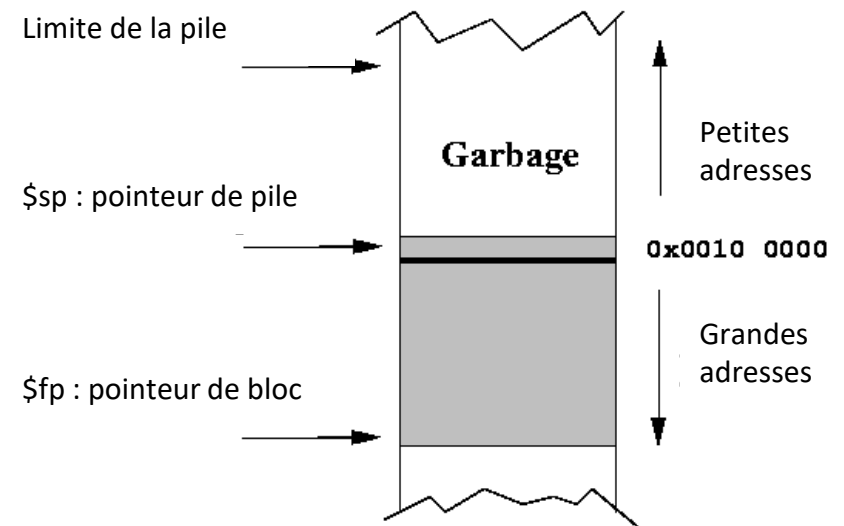


# Passage d'arguments

- ❑ Rappel : \$a0/\$a1/\$a2/\$a3
- ❑ Que faire pour passer 5 entiers ?
- ❑ Une chaîne de 16 caractères

# La pile

- ❑ Zone mémoire (RAM) tampon
- ❑ Bien plus de place que les registres
- ❑ Allocation dynamique
- ❑ Le fameux “stack overflow” la pile déborde sur un autre segment mémoire



# Offsets

- ❑ **spécifié en octets :  $n(\$t0)$**
- ❑ **revient à ajouter  $n$  octets à l'adresse**
  - ❑  $\$t0 \rightarrow 00000001$
  - ❑  $4(\$t0) \rightarrow 00100001$

# Utilisation de la pile

## ❑ Attention

- ❑ adressage à partir d'une grande adresse
- ❑ vous devez déplacer le pointeur vers le prochain espace libre

*#Push*

```
li $t3, 12
```

```
subi $sp, $sp, 4
```

```
sw $t3, 0($sp)
```

*#Pop*

```
lw $t3, 0($sp)
```

```
addi $sp, $sp, 4
```

```
li $v0, 1
```

```
mv $a0, $t3
```

```
syscall
```

# Utilisation de la pile

*#Push*

li \$t3, 12

subi \$sp, \$sp, 4

sw \$t3, 0(\$sp)

*#Pop*

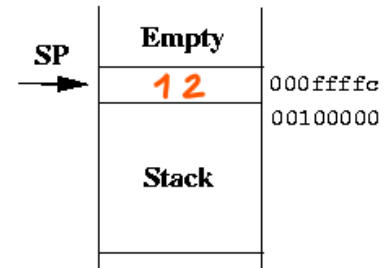
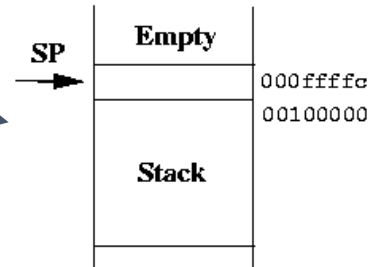
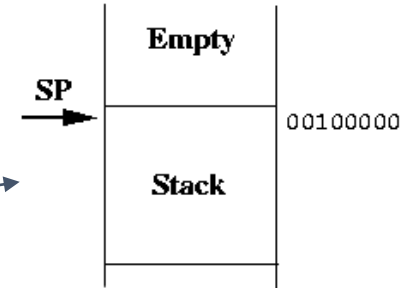
lw \$t3, 0(\$sp)

addi \$sp, \$sp, 4

li \$v0, 1

mv \$a0, \$t3

syscall



# Macros

```
.macro print_int (%x)
    li $v0, 1
    add $a0, $zero, %x
    syscall
.end_macro
```

- ❑ pas un “fonction”
- ❑ directive de réécriture
- ❑ + lisibilité du code
- ❑ + raccourcis les fichier sources