

Jeu d'instructions

TP 2

Objectifs

- ★ Familiarisation avec la structure d'un programme assembleur MIPS
- ★ Lire du code assembleur
- ★ Ecrire du code assembleur
- ★ Simuler du code assembleur

1 Variables

Exercice 1

1. Déclarez 4 octets *o1*, *o2*, *o3* et *o4* valant respectivement 1, 2, 3 et 4 puis un mot *m1* valant 0xAABBCCDD. Assemblez ce programme et regardez comment ces variables sont implantées en mémoire.
2. Donnez les adresses correspondant aux étiquettes du programme *o1*, *o2*, *o3*, *o4* et *m1*.
3. Dans l'onglet *Settings*, activez **Show Labels Window**. Qu'observez-vous ? Vérifiez ce que vous avez répondu à la question précédente.

2 Programmes avec accès mémoire

Exercice 2

1. Écrivez un programme assembleur comportant l'allocation aux adresses *v1* et *v2* de deux mots mémoire initialisés respectivement à -1 et 0xFF. Le programme doit charger les valeurs contenues aux adresses *v1* et *v2* dans les registres \$5 et \$6, puis afficher les deux valeurs. Assemblez et testez votre programme.
2. Modifiez le programme pour qu'il ajoute 1 à *v1* et à *v2* puis range les nouvelles valeurs en mémoire. Assemblez et exécutez le programme. Vérifiez le contenu de la mémoire à la fin de l'exécution (l'exécution a-t-elle bien modifié les valeurs implantées en mémoire aux adresses *v1* et *v2* ?).

Exercice 3

1. Écrivez un programme assembleur qui implante le programme C suivant :

```

int tab[] = {4, 23, 12, 3, 8, 1}

int main() {
    int s = tab[3];    /* s = 3 */
    int p = tab[4];    /* p = 8 */

    tab[0] = s + 1;
    tab[1] = s + p
    tab[2] = tab[5];

    return 0;
}

```

2. Exécutez votre programme pas à pas en regardant évoluer le contenu de la mémoire. Vérifiez que les valeurs contenues dans le tableau sont correctes à la fin de l'exécution.

Exercise 4

Remarques :

- On fait l'hypothèse que le type int en C est codé sur 4 octets (c'est généralement le cas sur les architectures 32 bits)
- Un tableau tab est un ensemble de valeurs rangées consécutivement en mémoire et tab[0] désigne le premier élément du tableau.
- Une chaîne de caractères ch est un tableau de caractères, déclarée comme un tableau, par exemple char ch[] = "exemple";.

1. Écrivez un programme assembleur correspondant au code C suivant :

```

int tab[] = { 1, 2, 34, 256, -1 }; /* tableau d'entiers */
char chaine[] = "toto";           /* chaîne de caractères */

int main() {
    printf("%d", tab[3]);
    printf("%d", chaine[2]);
    printf("%c", chaine[2]);
    return 0;
}

```

2. Quelles sont les valeurs affichées lors de l'exécution ?

3 Structure de contrôles

Exercise 5

Écrivez le code assembleur correspondant au programme C ci-dessous. Vous utiliserez le registre \$5 pour stocker la valeur de la variable a et le registre \$6 pour la valeur de la variable b.

```

int main() {
    int a = -5;
    int b = 3
    if (a == 0) {
        a = a + b;
    }
}

```

```
else {  
    a = a - b;  
}  
return 0;  
}
```

Exercise 6

Écrivez un programme assembleur correspondant au programme C suivant en mettant des commentaires pour donner la correspondance entre les variables locales du programme C et les registres que vous utilisez pour stocker le contenu de ces variables.

```
int main() {  
    int i;  
    int somme = 0;  
    for (i = 10; i > 0; i--) {  
        somme = somme + i;  
    }  
    printf("%d", somme);  
    return 0;  
}
```