

TP4 – Manipulation de tableaux

Objet du TP

Ce TP porte sur la manipulation de tableaux en assembleur Nios.

Démarrage

Comme pour le premier TP, un squelette de projet est disponible sur le *share*.

1. Copiez le répertoire **G:\I3info\tparc\TP5** dans votre espace personnel.

Important: vous rencontrerez des erreurs si vous omettez cette étape.

2. Lancez Altera Monitor en exécutant le fichier: **G:\I3info\tparc\monitor.bat**
3. Ouvrez le fichier **TP5\TP5.ncf** dans Altera Monitor.
4. Lorsqu'un dialogue vous propose de charger le processeur Nios II sur votre carte, vérifiez que celle-ci est branchée et allumée et appuyez sur **Oui**.

Travail attendu

Lecture / affichage d'un tableau

Vous allez écrire deux fonctions, *lectureTableau()* et *affichageTableau()*, dont le prototype est :

```
void lectureTableau(int *tableau, int taille);  
void affichageTableau(int *tableau, int taille);
```

Ces fonctions permettront respectivement de lire et d'afficher le contenu d'un tableau d'entiers, étant donné son adresse et le nombre d'éléments.

1. Complétez la fonction *main()* pour invoquer ces deux fonctions:

```
int tableau[10]; // Déclaré dans le segment de données  
void main() {  
    print_string("Lecture du tableau.\n");  
    lectureTableau(tableau, 10);  
    print_string("Affichage du tableau.\n");  
    affichageTableau(tableau, 10);  
    exit();  
}
```

Rappel: Par convention, les arguments sont passés via les registres **r4-r7**.

2. Implémentez la fonction *lectureTableau()* en assembleur:

```
void lectureTableau(int *tableau, int taille) {  
    int i;  
    for (i=0; i<taille; i++) {  
        print_string("Entrez un nombre:\n");  
        tableau[i] = read_int();  
    }  
}
```

Note: Selon les registres utilisés, utilisez la pile pour préserver leur valeur selon les conventions vues en cours.

3. Testez le bon fonctionnement de votre fonction en **inspectant la mémoire** après l'ajout de chaque élément.

Pour ce faire, déterminez l'adresse du tableau puis utilisez l'onglet *Memory* de la zone principale pour inspecter la mémoire à cette adresse. Vous pouvez passer en affichage décimal en sélectionnant **Number format > Decimal** dans le menu contextuel (clic-droit).

4. Implémentez la fonction *affichageTableau()* en assembleur:

```
void affichageTableau(int *tableau, int taille) {  
    int i;  
    for (i=0; i<taille; i++) {  
        print_int(tableau[i]);  
    }  
}
```

En exécutant votre programme, vérifiez que le tableau affiché est identique au tableau lu.

Inversion de tableau

1. Implémentez en assembleur la fonction suivante, permettant d'inverser le contenu d'un tableau :

```
void inversionTableau(int *tableau, int taille) {  
    int tmp;  
    int i=0, j=taille-1;
```

```
while (i<j) {  
    tmp = tableau[i];  
    tableau[i] = tableau[j];  
    tableau[j] = tmp;  
    i++; j--;  
}  
}
```

2. Modifiez votre programme pour invoquer cette fonction entre la lecture et l'affichage du tableau.
3. Vérifiez le bon fonctionnement de votre programme.

Compte-rendu

Vous adresserez un compte-rendu de votre travail.

Pensez à inclure la mention: ARC2 dans le sujet de votre mail.