TP4 ILM
Manipulation de structures

Objet du TP

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

Démarrage

1. Récupérez le dossier **TP4** auprès de votre encadrant.
2. Lancez Altera Monitor en exécutant le fichier : **share:\l3info\ILM\TP\monitor.bat**.
3. Ouvrez le fichier **TP4\TP4.amp** 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**.

Rendu

Le code sera évalué avec un rendu par mail à la fin du TP. Le mail contiendra ILM dans le sujet, avec le nom des deux personnes du binôme ainsi que leur groupe. Vous veillerez à la gestion de la pile, et les registres seront utilisés selon les conventions vues en cours.

Attention, seul le code commenté sera évalué !

Partie 1 : affichage d'une structure

Vous allez écrire une fonction, **`print_data()`** dont le prototype est :

```
struct data {
    int x;
    short y;
};
void print_data(struct data *a);
```

Cette fonction permet d'afficher le contenu d'une structure *data* étant donnée son adresse.

Question 1 :

Implémentez la fonction **`print_data()`** en assembleur. Vous utiliserez la pile pour sauver les valeurs des registres utilisés selon les conventions vues en cours.

```
void print_data(struct data *a) {
     printf("Valeur: %d, %d\n", a->x, a->y);
}
```

Question 2 :

Complétez la fonction **main()** pour invoquer cette fonction sur la structure **valeur** :

```
struct data valeur = { 14, 3 }; // Déclaré dans le segment de données
void main() {
     print_data(&valeur);
     exit();
}
```

Vérifiez son fonctionnement à l'exécution.

Partie 2 : Comparaison de deux structures

Question 1 :

Implémentez en assembleur la fonction suivante, permettant de comparer deux structures :

```
int compare_data(struct data *a, struct data *b) {
     if ((a->x == b->x) && (a->y == b->y))
          return 0;
     else if (a->x > b->x)
          return 1;
     else
          return -1;
}
```

Question 2 :

Complétez la fonction **main()** pour invoquer cette fonction sur la structure **valeur** et la case 2 de **tableau**, un tableau de structure **data** :

```
comp = compare_data(&valeur, &tableau[2]);
printf("%d", comp);
```

Vérifiez son fonctionnement à l'exécution.

Partie 3 : Recherche dichotomique sur un tableau de structure

Question 1 :

Modifiez la fonction **`rechercheDico()`** du TP3 pour utiliser la fonction **`compare_data()`** telle que définie dans la partie 2.

```
int rechercheDicho(struct data *val, struct data *tab, int debut,
int fin) {
     int pos, comp;
     if (debut > fin)
          return -1;
     pos = debut + (fin-debut) / 2;
     comp = compare_data(val, &tab[pos]);
     if(comp == 0)
          return pos;
     if (comp > 0)
          return rechercheDicho(val, tab, pos+1, fin);
     // comp < 0
     return rechercheDicho(val, tab, debut, pos-1);
}
```

Question 2 :

Complétez la fonction **`main()`** pour invoquer cette fonction sur la structure **`valeur`** et le **`tableau`** :

```
pos = rechercheDicho(&valeur, tableau, 0, 99);
if (pos < 0)
  printf("Nombre non trouve.\n");
else
  printf("La position du nombre est: %d\n", pos);
```

Vérifiez son fonctionnement à l'exécution.