**TP Architecture des ordinateurs**

**4 TP4 : Passage des arguments**

* étude d'une procédure d'échange de la valeur de deux entiers,
* passage des arguments par valeur,
* passage des arguments par pointeur.

**Mise en route**

* Lancez Turbo-C,
* Ouvrez le fichier EXCH.C.

**Le fichier Exch.c**

#include <stdio.h>

void exch( int n1, int n2 ) {

 int tmp;

        tmp = n1;

        n1 = n2;

        n2 = tmp;

 return; // fin de procedure

}

int nombre1, nombre2;

int main() {

        /\* passer rapidement en utilisant F4 \*/

        clrscr();

        scanf( "%d", &nombre1 );

        scanf( "%d", &nombre2 );

        /\* observer l'appel en utilisant F7 \*/

        exch( nombre1, nombre2 );

        printf( "Après\nnombre1 = %d\nnombre2 = %d\n", nombre1, nombre2 );

        return 0;

}

**Travail à réaliser**

Constatez l'inefficacité de cette procédure à échanger les valeurs de nombre1 et nombre2 en observant le déroulement de exch sous Turbo-Debugger.

* Relevez sur papier l'évolution de la pile et des registres. Faites apparaître le pointeur de pile SP et le pointeur de base de pile BP.
* Pourquoi les valeurs de nombre1 et nombre2 ne sont-elles pas échangées ?
* Qu'est-ce qui est échangé en réalité ?
* Comment modifier le programme pour que les valeurs de nombre1 et nombre2 soient bien échangées ?
* Continuez le TP avec EXCH1.C

**Le fichier Exch1.c**

#include <stdio.h>

#include <conio.h>

void **exch**( int far \*n1, int far \*n2 ) {

 int tmp;

 tmp = \*n1;

 \*n1 = \*n2;

 \*n2 = tmp;

 return; // fin de procedure

}

int nombre1, nombre2;

int main() {

 clrscr();

 printf( "Entrez un entier : " ); scanf( "%d", &nombre1 );

 printf( "Puis un autre entier : " ); scanf( "%d", &nombre2 );

 printf( "Avant\nnombre1 = %d\nnombre2 = %d\n", nombre1, nombre2 );

 **exch( &nombre1, &nombre2 );**

 printf( "Après\nnombre1 = %d\nnombre2 = %d\n", nombre1, nombre2 );

 return 0;

}

**Travail à réaliser**

Cette procédure échange bien nombre1 et nombre2.

* Relevez sur papier l'évolution de la pile et des registres. Faites apparaître le pointeur de pile SP et le pointeur de base de pile BP.
* Observez le code généré pour différentes options d'optimisation. Dans le Turbo-C, utilisez le menu Options / Compiler / Opimizations. Rubrique Register variables de la boîte de dialogue.
	1. pas d'optimisation, (choix None)
	2. optimisation par le mot-clé register appliqué à tmp, (choix Register keyword)
	3. optimisation automatique (choix Automatic).
* Quel couple de registres est utilisé par le compilateur pour les pointeurs far ?
* Quelle est l'adresse segmentée complète de nombre1 et nombre2 ?
* Continuez le TP avec EXCH2.C

**Le fichier Exch2.c**

#include <stdio.h>

void exch( int far \*n1, int far \*n2 ) {

 return; // fin de procedure

}

int nombre1, nombre2;

int main() {

 clrscr();

 printf( "Entrez un entier : " ); scanf( "%d", &nombre1 );

 printf( "Puis un autre entier : " ); scanf( "%d", &nombre2 );

 printf( "Avant\nnombre1 = %d\nnombre2 = %d\n", nombre1, nombre2 );

 exch( &nombre1, &nombre2 );

 printf( "Après\nnombre1 = %d\nnombre2 = %d\n", nombre1, nombre2 );

 return 0;

}

**Travail à réaliser**

Codez la procédure exch en assembleur de façon plus efficace que le compilateur, c'est-à-dire avec moins d'instructions ou des instructions moins coûteuses.