

EXERCICE 4_2 PIC32MX

OBJECTIFS

Cet exercice a pour objectif de permettre aux étudiants de découvrir comment sont réalisés les passages de paramètres et comment est gérée la pile lors des appels aux fonctions.

Ceci en utilisant la possibilité du MPLAB X de fournir un listing assembleur.

PREPARATION

Création d'un projet Standalone pour le PIC32MX795F512L, à nommer Ex4_2. Ajout du fichier sources main_Ex4_2.c qui se trouve sous :

...\Maitres-Eleves\SLO\Modules\SL229_MINF\...\Ex4_2

FICHER SOURCE C

```
// Fonction AddAB
int AddAB (int A, int B)
{
    return (A + B);
}

// FonctionA
int FunctionA (int ValA, int ValB)
{
    int Res;
    Res = AddAB(ValA, ValB);
    return(Res);
}

void main() {
    int Res1, Res2;
    int N1 = 10;
    int N2 = 20;
    int N3 = 30;
    int N4 = 40;

    // Appel de la fonction FunctionA
    Res1 = FunctionA(N1, N2);
    Res2 = FunctionA(N3, N4);

    while(1);
}
```

LISTING ASSEMBLEUR (COMPILATEUR XC32 V2.15)

Disassembly Listing for Mc32Ex_4_2

Generated From:

C:/microchip/harmony/v2_06/apps/MINF/Exercices/Mc32Ex_4_2/Mc32Ex_4_2.X/dist/default/production/Mc32Ex_4_2.X.production.elf

27 nov. 2019 10:37:46

c:/microchip/harmony/v2_06/apps/minf/exercices/mc32ex_4_2/main_ex4_2.c -----

```

1:                                     #include <stddef.h>
// Defines NULL
2:                                     #include <stdbool.h>
// Defines true
3:                                     #include <stdlib.h>
// Defines EXIT_FAILURE
4:                                     #include <stdint.h>
// Defines EXIT_FAILURE
5:
6:                                     // Fonction AddAB
7:                                     int AddAB (int A, int B)
8:                                     {
9D000000  27BDFFF8  ADDIU SP, SP, -8
9D000004  AFBE0004  SW FP, 4(SP)
9D000008  03A0F021  ADDU FP, SP, ZERO
9D00000C  AFC40008  SW A0, 8(FP)
9D000010  AFC5000C  SW A1, 12(FP)
9:                                     return (A + B);
9D000014  8FC30008  LW V1, 8(FP)
9D000018  8FC2000C  LW V0, 12(FP)
9D00001C  00621021  ADDU V0, V1, V0
10:                                    }
9D000020  03C0E821  ADDU SP, FP, ZERO
9D000024  8FBE0004  LW FP, 4(SP)
9D000028  27BD0008  ADDIU SP, SP, 8
9D00002C  03E00008  JR RA
9D000030  00000000  NOP
11:
12:                                     // FonctionA
13:                                     int FunctionA (int ValA, int ValB)
14:                                     {
9D000034  27BDFFE0  ADDIU SP, SP, -32
9D000038  AFBF001C  SW RA, 28(SP)
9D00003C  AFBE0018  SW FP, 24(SP)
9D000040  03A0F021  ADDU FP, SP, ZERO
9D000044  AFC40020  SW A0, 32(FP)
9D000048  AFC50024  SW A1, 36(FP)
15:                                     int Res;
16:                                     Res = AddAB(ValA, ValB);
9D00004C  8FC40020  LW A0, 32(FP)

```

```

9D000050  8FC50024  LW A1, 36(FP)
9D000054  0F400000  JAL AddAB
9D000058  00000000  NOP
9D00005C  AFC20010  SW V0, 16(FP)
17:                                     return(Res);
9D000060  8FC20010  LW V0, 16(FP)
18:                                     }
9D000064  03C0E821  ADDU SP, FP, ZERO
9D000068  8FBF001C  LW RA, 28(SP)
9D00006C  8FBE0018  LW FP, 24(SP)
9D000070  27BD0020  ADDIU SP, SP, 32
9D000074  03E00008  JR RA
9D000078  00000000  NOP
19:
20:                                     void main() {
9D00007C  27BDFFD0  ADDIU SP, SP, -48
9D000080  AFBF002C  SW RA, 44(SP)
9D000084  AFBE0028  SW FP, 40(SP)
9D000088  03A0F021  ADDU FP, SP, ZERO
21:                                     int Res1, Res2;
22:                                     int N1 = 10;
9D00008C  2402000A  ADDIU V0, ZERO, 10
9D000090  AFC20010  SW V0, 16(FP)
23:                                     int N2 = 20;
9D000094  24020014  ADDIU V0, ZERO, 20
9D000098  AFC20014  SW V0, 20(FP)
24:                                     int N3 = 30;
9D00009C  2402001E  ADDIU V0, ZERO, 30
9D0000A0  AFC20018  SW V0, 24(FP)
25:                                     int N4 = 40;
9D0000A4  24020028  ADDIU V0, ZERO, 40
9D0000A8  AFC2001C  SW V0, 28(FP)
26:
27:                                     // Appel de la fonction FunctionA
28:                                     Res1 = FunctionA(N1, N2);
9D0000AC  8FC40010  LW A0, 16(FP)
9D0000B0  8FC50014  LW A1, 20(FP)
9D0000B4  0F40000D  JAL FunctionA
9D0000B8  00000000  NOP
9D0000BC  AFC20020  SW V0, 32(FP)
29:                                     Res2 = FunctionA(N3, N4);
9D0000C0  8FC40018  LW A0, 24(FP)
9D0000C4  8FC5001C  LW A1, 28(FP)
9D0000C8  0F40000D  JAL FunctionA
9D0000CC  00000000  NOP
9D0000D0  AFC20024  SW V0, 36(FP)
30:
31:                                     while(1);
9D0000D4  0B400035  J 0x9D0000D4
32:                                     }
    
```

Il vous est demandé de compléter l'évolution de la pile lors du 1^{er} appel de la FonctionA, situation lorsqu'on se trouve dans AddAB.

Dans AddAB

[illegible]