

2. SOUS-PROGRAMMES ASSEMBLEUR. Interface avec le langage C

Interface avec le langage C

Support physique pour le passage de paramètres

- . Transmission des paramètres aux fonctions par la pile.
- . Retour de la fonction asm vers le langage évolué dans EAX.

APPEL D'UNE FONCTION C PAR UNE ROUTINE ASM

```
// Fichier AdditionC.c
// APPEL D'UNE FONCTION C PAR UNE ROUTINE ASM
// La fonction C effectue une addition

#include <stdio.h>
int SOM (int x, int y);      // Declaration fonction C

int SOM (int x, int y)      // Routine C d'Addition
{ // La fonction C va chercher ses parametres dans la pile
  return (x+y); // La fonction C place le resultat dans eax
}

void main()
{ int somme, a, b;
  scanf ("%d", &a);printf("\n b = ? ");  scanf ("%d", &b);

  __asm
  {
    push b;           // Empilement de b
    push a;           // Empilement de a
    callSOM;          // Appel (a + b); Resultat tjs dans eax
    mov somme, eax    // Memorisation du resultat
  }
  printf("\nSomme = %d\n\n", somme);
}
```

APPEL D'UNE FONCTION ASM PAR UN PGM C*// Fichier AdditionASM.c**// APPEL D'UNE PROCEDURE ASM PAR UN PGM C**// La procedure ASM effectue une addition*

#include <stdio.h>

int ADDITION (int a, int b); // Declaration fonction ASM

void main()

{

int somme;

int x, y;

printf("\n x = ? "); scanf ("%d", &x);

printf("\n y = ? "); scanf ("%d", &y);

somme = ADDITION (x, y);

printf("\nsomme = %d\n\n", somme);

}

int ADDITION (int a, int b)

{

// Routine ASM d'Addition

asm

{

mov eax, a; ; 1er argument reference par son nom

mov ecx, b; ; 2nd argument reference par son nom

add eax, ecx ; le resultat doit tjs etre place dans eax

}

}