

nov. 06, 20 13:48 **max_v5.s** Page 1/3

```

.global a
.global b
.global max_ab
.global c
.global d
.global max_cd

.data
a:      .word 3      @ unsigned int a = 3
b:      .word 5      @ unsigned int b = 5
c:      .word 6      @ unsigned int c = 6
d:      .word 4      @ unsigned int d = 4

.bss
max_ab: .skip 4      @ unsigned int max_ab
max_cd: .skip 4      @ unsigned int max_cd

        @ pour sauvegarde des registres dans calcul
CALCUL_DELTA_LR = -0
CALCUL_DELTA_R6 = -4
CALCUL_DELTA_R5 = -8
CALCUL_DELTA_R4 = -12
CALCUL_DELTA_R0 = -16

        .skip 4      @ r0
        .skip 4      @ r4
        .skip 4      @ r5
        .skip 4      @ r6
local_calcul: .skip 4 @ lr

        @ pour sauvegarde des registres dans max2
MAX2_DELTA_R0 = -12
MAX2_DELTA_R1 = -8
MAX2_DELTA_R2 = -4
MAX2_DELTA_R8 = -0
MAX2_DELTA_X = 4
MAX2_DELTA_Y = 8
MAX2_DELTA_MAXI = 12

        .skip 4      @ r0
        .skip 4      @ r1
        .skip 4      @ r2
local_max2: .skip 4   @ r8
        .skip 4      @ structure des parametres reçus : x
        .skip 4      @ y
        .skip 4      @ maxi

.global calcul

.text

@ Convention d'appel de max2 : x,y,maxi dans local_max2
@ r0 : copie de x  r1 : copie de y r2 : copie de maxi
@
@ Variable locale rmax : r8

max2:      @ prologue : sauvegarde de r8
        ldr fp,=local_max2
        str r8,[fp,#MAX2_DELTA_R8]
        str r2,[fp,#MAX2_DELTA_R2]
        str r1,[fp,#MAX2_DELTA_R1]

```

nov. 06, 20 13:48 **max_v5.s** Page 2/3

```

        str r0,[fp,#MAX2_DELTA_R0]

        ldr r0,[fp,#MAX2_DELTA_X]      @ *&x
        ldr r1,[fp,#MAX2_DELTA_Y]      @ *&y
        ldr r2,[fp,#MAX2_DELTA_MAXI]   @ *&maxi

        @ corps inchangé
        mov r8,r0                      @ rmax=0

if1:      cmp r1,r8                    @ if (y ≤ rmax) goto finsil
        ble finsil

alors1:   mov r8,r1                    @ rmax = y
finsil:   str r8,[r2]                  @ **&maxi = rmax

        @ epilogue : restauration et retour
        ldr r0,[fp,#MAX2_DELTA_R0]
        ldr r1,[fp,#MAX2_DELTA_R1]
        ldr r2,[fp,#MAX2_DELTA_R2]
        ldr r8,[fp,#MAX2_DELTA_R8]
        mov pc,lr

@ Convention d'appel de calcul
@ adresse de retour : lr
@
@ Variables locales e : r4  f: r5  max_ef : r6
@ temporaire : r0

calcul :      @ prologue : sauvegarder les registres
        ldr fp,=local_calcul
        str lr,[fp,#CALCUL_DELTA_LR]
        str r6,[fp,#CALCUL_DELTA_R6]
        str r5,[fp,#CALCUL_DELTA_R5]
        str r4,[fp,#CALCUL_DELTA_R4]
        str r0,[fp,#CALCUL_DELTA_R0]

        mov r4,#5                      @ r4=5
        mov r5,#6                      @ r5=6

        mov r6,r4                      @ max_ef=e

if2:      cmp r5,r6                    @ if (f ≤ max_ef) gotofinsi2
        ble finsi2

alors2:   mov r6,r5                    @ max_ef = f
finsi2:   @ max2(a,b,&max_ab)

        ldr sp,=local_max2
        ldr r0,=a                      @ x = a
        ldr r0,[r0]
        str r0,[sp, #MAX2_DELTA_X]

        ldr r0,=b                      @ y = b
        ldr r0,[r0]
        str r0,[sp, #MAX2_DELTA_Y]

        ldr r0,=max_ab                 @ maxi &max_ab
        str r0,[sp, #MAX2_DELTA_MAXI]

        bl max2

        @ max2(a,b,&max_ab)

```

nov. 06, 20 13:48

max_v5.s

Page 3/3

```
ldr  r0,=c          @ x = c
ldr  r0,[r0]
str  r0,[sp, #MAX2_DELTA_X]

ldr  r0,=d          @ y = d
ldr  r0,[r0]
str  r0,[sp, #MAX2_DELTA_Y]

ldr  r0,=max_cd     @ maxi = &max_cd
str  r0,[sp, #MAX2_DELTA_MAXI]

bl   max2

      @ epilogue : restaurer lr + retour
ldr  fp,=local_calcul
ldr  r0,[fp,#CALCUL_DELTA_R0]
ldr  r4,[fp,#CALCUL_DELTA_R4]
ldr  r5,[fp,#CALCUL_DELTA_R5]
ldr  r6,[fp,#CALCUL_DELTA_R6]
ldr  lr,[fp,#CALCUL_DELTA_LR]
mov  pc,lr

.ltorg
```