

oct. 24, 18 17:02	exemple_proc_simple.s.7bis.latin.txt	Page 1/3
<pre> int y = 2; int t; void gm (void) void m (int x) void f (int a, int b, int c) { int z; { z = 3; { f (3,4,5); int vloc; int y; /* différent de y global */ m (2); int v; y = a + b; z++; f(x+2,y,3); t = y + c; f (1,0,2*z); vloc++; } z = z + 4; } } .global main y: .data .word 2 t: .bss .skip 4 privee_f: .skip 4 @ variable locale y -16 .skip 4 @ sauvegarde r1 -12 .skip 4 @ sauvegarde r0 -8 .skip 4 @ sauvegarde fp -4 param_f: .skip 4 @ parametre a +0 .skip 4 @ parametre b +4 .skip 4 @ parametre c +8 DELTA_F_Y = -16 DELTA_F_SAUUV_R1 = -12 DELTA_F_SAUUV_R0 = -8 DELTA_F_SAUUV_FP = -4 DELTA_F_A = 0 DELTA_F_B = 4 DELTA_F_C = 8 privee_m: .skip 4 @ variable locale vloc -24 .skip 4 @ variable locale v -20 .skip 4 @ sauvegarde r0 -16 .skip 4 @ sauvegarde fp -12 .skip 4 @ sauvegarde sp -8 .skip 4 @ sauvegarde lr/@sse retour -4 param_m: .skip 4 @ parametre x +0 DELTA_M_VLOC = -24 DELTA_M_V = -20 DELTA_M_SAUUV_R0 = -16 DELTA_M_SAUUV_FP = -12 DELTA_M_SAUUV_SP = -8 DELTA_M_SAUUV_LR = -4 DELTA_M_X = 0 privee_gm: .skip 4 @ variable locale z -20 .skip 4 @ sauvegarde r0 -16 .skip 4 @ sauvegarde fp -12 .skip 4 @ sauvegarde sp -8 .skip 4 @ sauvegarde lr/@sse retour -4 param_gm: </pre>		

oct. 24, 18 17:02	exemple_proc_simple.s.7bis.latin.txt	Page 2/3
<pre> DELTA_GM_Z = -20 DELTA_GM_SAUUV_R0 = -16 DELTA_GM_SAUUV_FP = -12 DELTA_GM_SAUUV_SP = -8 DELTA_GM_SAUUV_LR = -4 .text @ convention : les procédures utilisent sp pour remplir les paramètres @ passés aux procédures qu'elles appellent et fp pour @ accéder aux paramètres qu'elles ont reçu et à leurs @ variables locales. @ le registre ip/r12 n'est pas sauvegardé lors d'un appel main: bl gm mov pc, lr gm: ldr ip, =param_gm @ on utilise ip pour sauver fp str fp, [ip, #DELTA_GM_SAUUV_FP] mov fp, ip str lr, [fp, #DELTA_GM_SAUUV_LR] str sp, [fp, #DELTA_GM_SAUUV_SP] str r0, [fp, #DELTA_GM_SAUUV_R0] mov r0, #3 @ z = 3 str r0, [fp, #DELTA_GM_Z] ldr sp, =param_f @ f(a=3,b=4,c=5) mov r0, #3 str r0, [sp, #DELTA_F_A] mov r0, #4 str r0, [sp, #DELTA_F_B] mov r0, #5 str r0, [sp, #DELTA_F_C] bl f ldr sp, =param_m @ m(x=2) mov r0, #2 str r0, [sp, #DELTA_M_X] bl m ldr r0, [fp, #DELTA_GM_Z] @ z++ add r0, r0, #1 str r0, [fp, #DELTA_GM_Z] ldr sp, =param_f @ f(a=1, b=0, c=2*z) mov r0, #1 str r0, [sp, #DELTA_F_A] mov r0, #0 str r0, [sp, #DELTA_F_B] ldr r0, [fp, #DELTA_GM_Z] add r0, r0, r0 str r0, [sp, #DELTA_F_C] bl f ldr r0, [fp, #DELTA_GM_Z] @ z=z+4 add r0, r0, #4 str r0, [fp, #DELTA_GM_Z] ldr r0, [fp, #DELTA_GM_SAUUV_R0] @ restaurer ldr sp, [fp, #DELTA_GM_SAUUV_SP] ldr lr, [fp, #DELTA_GM_SAUUV_LR] </pre>		

oct. 24, 18 17:02

exemple_proc_simple.s.7bis.latin.txt

Page 3/3

```

        ldr    fp, [fp, #DELTA_GM_SAUV_FP]
        mov    pc, lr                @ retour

m:      ldr    ip, = param_m          @ on utilise ip pour sauver fp
        str    fp, [ip, #DELTA_M_SAUV_FP]
        mov    fp, ip
        str    lr, [fp, #DELTA_M_SAUV_LR]
        str    sp, [fp, #DELTA_M_SAUV_SP]
        str    r0, [fp, #DELTA_M_SAUV_R0]

        ldr    sp, =param_f          @ f(a=x+2, b=y, c=5)
        ldr    r0, [fp, #DELTA_M_X]
        add    r0, r0, #2
        str    r0, [sp, #DELTA_F_A]
        ldr    r0, =y
        ldr    r0, [r0]
        str    r0, [sp, #DELTA_F_B]
        mov    r0, #3
        str    r0, [sp, #DELTA_F_C]
        bl     f

        ldr    r0, [fp, #DELTA_M_VLOC] @ vloc++
        add    r0, r0, #1
        str    r0, [fp, #DELTA_M_VLOC]

        ldr    lr, [fp, #DELTA_M_SAUV_R0] @ restaurer
        ldr    sp, [fp, #DELTA_M_SAUV_SP]
        ldr    lr, [fp, #DELTA_M_SAUV_LR]
        ldr    fp, [fp, #DELTA_M_SAUV_FP]
        mov    pc, lr                @ retour

f:      ldr    ip, =param_f          @ on utilise ip pour sauver fp
        str    fp, [ip, #DELTA_F_SAUV_FP]
        mov    fp, ip
        str    r0, [fp, #DELTA_F_SAUV_R0]
        str    r1, [fp, #DELTA_F_SAUV_R1]

        ldr    r0, [fp, #DELTA_F_A]    @ y = a + b
        ldr    r1, [fp, #DELTA_F_B]
        add    r0, r0, r1
        str    r0, [fp, #DELTA_F_Y]

        ldr    r0, [fp, #DELTA_F_Y]    @ t = y + c
        ldr    r1, [fp, #DELTA_F_C]
        add    r0, r0, r1
        ldr    r1, =t
        str    r0, [r1]

        ldr    r1, [fp, #DELTA_F_SAUV_R1] @restaurer
        ldr    r0, [fp, #DELTA_F_SAUV_R0]
        ldr    fp, [fp, #DELTA_F_SAUV_FP]
        mov    pc, lr                @ retour

```