Rendu 2: Compilation

Maxime BRASLEY / Alexis LOPES VAZ / Gabin HEMMERLE / Hawa DIABATE

Partie 1:

```
Code Assembleur Obtenu
              Code Assembleur Attendu
                                                 | Code genere par le compilateur
Arbre 1:
        CMOVE(pile,SP)
                                                 .include beta.uasm
        BR(debut)
                                                 .include intio.uasm
                                                 .options tty
PUSH (LP)
PUSH (BP)
                                                     CMOVE (pile, SP)
MOVE (SP, BP)
                                                     BR (debut)
ALLOCATE (0)
DEALLOCATE (0)
                                                 main:
POP (BP)
POP (LP)
                                                 PUSH (LP)
RTN()
                                                 PUSH (BP)
                                                 MOVE (SP, BP)
debut:
                                                 ALLOCATE (0)
CALL(main)
HALT()
                                                 DEALLOCATE (0)
pile:
                                                 POP (BP)
                                                 POP (LP)
                                                 RTN()
                                                 debut:
                                                 CALL(main)
                                                 HALT()
                                                 pile:
                                                  Test #1
                                                  nom : main; type : void; categorie : fonction;
Arbre 2:
                                                   Code genere par le compilateur
        CMOVE(pile,SP)
                                                 .include beta.uasm
       BR(debut)
                                                 .include intio.uasm
i: LONG(10)
                                                 .options tty
j: LONG(20)
k: LONG(0)
I: LONG(0)
                                                     CMOVE (pile, SP)
                                                     BR (debut)
main:
                                                 i: LONG(10)
PUSH (LP)
                                                 j: LONG(20)
PUSH (BP)
MOVE (SP, BP)
                                                 k: LONG(0)
ALLOCATE (0)
                                                 1: LONG(0)
DEALLOCATE (0)
POP (BP)
POP (LP)
                                                 main:
                                                 PUSH (LP)
RTN()
                                                 PUSH (BP)
debut:
                                                 MOVE (SP, BP)
CALL(main)
                                                 ALLOCATE (0)
HALT()
                                                 DEALLOCATE (0)
pile:
```

```
POP (BP)
                                                   POP (LP)
                                                   RTN()
                                                   debut:
                                                   CALL(main)
                                                   HALT ()
                                                   pile:
                                                    Test #2
                                                    nom : main; type : void; categorie : fonction;
                                                    nom : i; type : int; categorie : global; valeur : 10;
                                                    nom : j; type : int; categorie : global; valeur : 20;
                                                     nom : k; type : int; categorie : global;
                                                     nom : l; type : int; categorie : global;
Arbre 3:
                                                    | Code genere par le compilateur
        CMOVE(pile,SP)
                                                   .include beta.uasm
        BR(debut)
                                                   .include intio.uasm
i: LONG(10)
j: LONG(20)
                                                   .options tty
k: LONG(0)
I: LONG(0)
                                                       CMOVE (pile, SP)
                                                       BR (debut)
main:
                                                   i: LONG(10)
PUSH (LP)
PUSH (BP)
                                                   j: LONG(20)
MOVE (SP, BP)
                                                   k: LONG(0)
ALLOCATE (0)
                                                   1: LONG(0)
CMOVE(2, r0)
                                                   main:
PUSH (r0)
                                                   PUSH (LP)
POP (r0)
                                                   PUSH (BP)
ST(r0, k)
                                                   MOVE (SP, BP)
                                                   ALLOCATE (0)
                                                   CMOVE(2, r0)
LD(i, r0)
PUSH (r0)
                                                   PUSH (r0)
                                                   POP (r0)
                                                   ST(r0, k)
LD(j, r0)
                                                   LD(i, r0)
PUSH (r0)
                                                   PUSH (r0)
                                                   LD(j, r0)
CMOVE(3, r0)
                                                   PUSH (r0)
PUSH (r0)
                                                   CMOVE(3, r0)
                                                   PUSH (r0)
mul 3 j
POP (r2)
                                                   POP (r2)
POP (r1)
                                                   POP (r1)
MUL (r1, r2, r3)
                                                   MUL (r1, r2, r3)
PUSH (r3)
                                                   PUSH (r3)
addition
                                                   POP (r2)
POP (r2)
                                                   POP (r1)
POP (r1)
                                                   ADD (r1, r2, r3)
ADD (r1, r2, r3)
                                                   PUSH (r3)
PUSH (r3)
                                                   POP (r0)
                                                   ST(r0, 1)
POP (r0)
                                                   DEALLOCATE (0)
ST(r0, I)
                                                   POP (BP)
DEALLOCATE (0)
                                                   POP (LP)
POP (BP)
POP (LP)
                                                   RTN()
RTN()
```

```
debut:
                                                     debut:
CALL(main)
                                                     CALL(main)
HALT()
                                                     HALT()
pile:
                                                    pile:
                                                     nom : main; type : void; categorie : fonction;
                                                     nom : i; type : int; categorie : global; valeur : 10;
                                                     nom : j; type : int; categorie : global; valeur : 20;
                                                     nom : k; type : int; categorie : global;
                                                     nom : l; type : int; categorie : global;
Arbre 4:
                                                     | Code genere par le compilateur
        CMOVE(pile,SP)
                                                     .include beta.uasm
        BR(debut)
                                                     .include intio.uasm
i: LONG(0)
                                                     .options tty
j: LONG(20)
main:
                                                        CMOVE (pile, SP)
PUSH (LP)
                                                        BR (debut)
PUSH (BP)
                                                    i: LONG(0)
MOVE (SP, BP)
                                                     j: \overline{\text{LONG}(20)}
ALLOCATE (0)
llis i
                                                    main:
RDINT()
                                                    PUSH (LP)
PUSH (r0)
                                                    PUSH (BP)
POP (r0)
ST(r0, i)
                                                    MOVE (SP, BP)
                                                    ALLOCATE (0)
                                                    RDINT()
LD(i, r0)
                                                    PUSH (r0)
PUSH (r0)
                                                    POP (r0)
LD(j, r0)
PUSH (r0)
                                                     ST(r0, i)
POP (r2)
                                                    LD(i, r0)
POP (r1)
                                                    PUSH (r0)
ADD (r1, r2, r3)
                                                    LD(j, r0)
PUSH (r0)
PUSH (r3)
|ecrire
                                                    POP (r2)
POP (r0)
                                                    POP (r1)
WRINT()
                                                    ADD (r1, r2, r3)
DEALLOCATE (0)
POP (BP)
                                                    PUSH (r3)
POP (LP)
                                                    POP (r0)
RTN()
                                                    WRINT ()
                                                    DEALLOCATE (0)
debut:
                                                    POP (BP)
CALL(main)
                                                     POP (LP)
HALT()
pile:
                                                    RTN()
                                                     debut:
                                                     CALL(main)
                                                    HALT ()
                                                    pile:
                                                     nom : i; type : int; categorie : global;
                                                     nom : j; type : int; categorie : global; valeur : 20;
                                                     nom : main; type : void; categorie : fonction;
                                                      Code genere par le compilateur
Arbre 5:
        CMOVE(pile,SP)
                                                     .include beta.uasm
        BR(debut)
                                                     .include intio.uasm
i: LONG(0)
```

```
options tty
main.
PUSH (LP)
                                                      CMOVE (pile, SP)
PUSH (BP)
                                                      BR (debut)
MOVE (SP, BP)
ALLOCATE (0)
                                                  i: LONG(0)
                                                  main:
RDINT()
                                                  PUSH (LP)
PUSH (r0)
POP (r0)
                                                  PUSH (BP)
ST(r0, i)
                                                  MOVE (SP, BP)
                                                  ALLOCATE (0)
si:
                                                  RDINT()
|comparaison
                                                  PUSH (r0)
LD(i, r0)
PUSH (r0)
                                                  POP (r0)
CMOVE(10, r0)
                                                  ST(r0, i)
PUSH (r0)
                                                  si_1:
POP (r2)
                                                  LD(i, r0)
POP (r1)
                                                  PUSH (r0)
CMPLT (r1, r2, r3)
PUSH (r3)
                                                  CMOVE(10, r0)
POP (r0)
                                                  PUSH (r0)
BF(R0, sinon)
                                                  POP (r2)
                                                  POP (r1)
alors:
lecris 1
                                                  CMPLT (r1, r2, r3)
CMOVE(1, r0)
                                                  PUSH (r3)
PUSH (r0)
                                                  POP (r0)
POP (r0)
WRINT()
                                                 BF(R0, sinon 1)
BR(fsi)
                                                  alors 1:
                                                  CMOVE(1, r0)
sinon:
                                                  PUSH (r0)
ecris 2
                                                  POP (r0)
CMOVE(2, r0)
                                                  WRINT()
PUSH (r0)
POP (r0)
                                                  BR(fsi 1)
WRINT()
                                                  sinon 1:
                                                  CMOVE(2, r0)
fsi:
                                                  PUSH (r0)
DEALLOCATE (0)
POP (BP)
                                                  POP (r0)
POP (LP)
                                                  WRINT()
RTN()
                                                  fsi 1:
                                                  DEALLOCATE (0)
debut:
                                                  POP (BP)
CALL(main)
HALT()
                                                  POP (LP)
pile:
                                                  RTN()
                                                  debut:
                                                  CALL(main)
                                                  HALT()
                                                  pile:
                                                   nom : i; type : int; categorie : global;
                                                   nom : main; type : void; categorie : fonction;
Arbre 6:
                                                   | Code genere par le compilateur
                                                  .include beta.uasm
CMOVE(pile,SP)
                                                  .include intio.uasm
BR(debut)
                                                  .options tty
data
                                                      CMOVE (pile, SP)
i:LONG(0)
                                                      BR (debut)
n:LONG(5)
                                                  i: LONG(0)
```

```
instructions
                                                  h: LONG(5)
main:
                                                  main:
    PUSH(LP)
                                                  PUSH (LP)
    PUSH(BP)
    MOVE(SP,BP)
                                                  PUSH (BP)
    ALLOCATE(0)
                                                  MOVE (SP, BP)
    CMOVE(0, R0)
                                                  ALLOCATE (0)
    PUSH(R0)
                                                  CMOVE(0, r0)
    POP(R0)
    ST(R0, i)
                                                  PUSH (r0)
                 BR(tq)
                                                  POP (r0)
 tq:
                                                  ST(r0, i)
    LD(i, R0)
                                                  tantque 1:
    PUSH(R0)
                                                  LD(i, r0)
    LD(n,R0)
    PUSH(R0)
                                                  PUSH (r0)
    POP(R1)
                                                  LD(h, r0)
    POP(R2)
                                                  PUSH (r0)
    CMPLT(R2,R1,R0)
                                                  POP (r2)
    PUSH(R0)
                                                  POP (r1)
    POP(R0)
    BF(R0, ftq_1)
                                                  CMPLT (r2, r1, r3)
                 BR(alors)
                                                  PUSH (r3)
alors:
                                                  POP (R0)
    PUSH(R0)
                                                  BF(R0, ftantque_1)
    POP(R0)
    WRINT()
                                                  LD(i, r0)
    LD(i, R0)
                                                  PUSH (r0)
    PUSH(R0)
                                                  POP (r0)
    CMOVE(1, R0)
                                                  WRINT()
    PUSH(R0)
    POP(R2)
                                                  LD(i, r0)
    POP(R1)
                                                  PUSH (r0)
    ADD(R2,R1,R0)
                                                  CMOVE(1, r0)
    PUSH(R0)
                                                  PUSH (r0)
    POP(R0)
                                                  POP (r2)
    ST(RO, i)
                  BR(tq)
                                                  POP (r1)
ftq_1:
                                                  ADD (r1, r2, r3)
    BR(ret_main)
                                                  PUSH (r3)
ret main :
                                                  POP (r0)
    DEALLOCATE(0)
    POP(BP)
                                                  ST(r0, i)
    POP(LP)
                                                  BR(tantque 1)
    RTN()
                                                  ftantque 1:
debut:
                                                  DEALLOCATE (0)
    CALL(main)
                                                  POP (BP)
    HALT()
     pile:
                                                  POP (LP)
                                                  RTN()
                                                  debut:
                                                  CALL(main)
                                                  HALT()
                                                  pile:
                                                   nom : i; type : int; categorie : global;
                                                   nom : h; type : int; categorie : global; valeur : 5;
                                                   nom : main; type : void; categorie : fonction;
Arbre 7:
                                                    Code genere par le compilateur
CMOVE(PILE,SP)
                                                   include beta.uasm
BR(debut)
                                                  .options tty
|data
                                                     CMOVE(pile,SP)
                                                     BR (debut)
Main:
                                                  a: LONG(10)
PUSH(LP)
PUSH(BP)
```

```
MOVE(SP,BP)
                                                                 PUSH (LP)
ALLOCATE(0)
                                                                 PUSH (BP)
                                                                 MOVE (SP, BP)
CMOVE(3,R0)
                                                                 ALLOCATE (2)
                                                                 CMOVE(1, r0)
PUSH (r0)
PUSH(R0)
CALL(f)
DEALLOCATE(1)
                                                                 POP (r0)
                                                                 PUTFRAME(r0, 1)
                                                                 CMOVE(1, r0)
LD(a,R0)
                                                                 PUSH (r0)
PUSH(R0)
                                                                 POP (r0)
POP(R0)
                                                                 PUTFRAME(r0, 5)
WRINT(R0)
                                                                 GETFRAME(r0, 1)
POP(LP)
                                                                 PUSH (r0)
POP(BP)
                                                                 GETFRAME(r0, 1)
DEALLOCATE(0)
                                                                 PUSH (r0)
RTN
                                                                 GETFRAME(r0, 5)
                                                                 PUSH (r0)
POP (r2)
f:
                                                                 POP (r1)
PUSH(LP)
                                                                 ADD (r1, r2, r3)
PUSH (r3)
PUSH(BP)
CMOVE(LP,BP)
                                                                 POP (r2)
POP (r1)
ALLOCATE(2)
                                                                 ADD (r1, r2, r3)
PUSH (r3)
|x=1
                                                                 POP (r0)
CMOVE(1,R0)
                                                                 ST(r0, a)
DEALLOCATE (2)
PUSH(R0)
POP(R0)
                                                                 POP (BP)
PUTFRAME(R0,0)
                                                                 POP (LP)
                                                                 RTN()
CMOVE(1,R0)
PUSH(R0)
                                                                 main:
POP(R0)
                                                                 PUSH (LP)
PUTFRAME(R0,4)
                                                                 PUSH (BP)
MOVE (SP, BP)
GETFRAME(0,R0)
PUSH(R0)
                                                                 ALLOCATE (0)
GETFRAME(4,R0)
                                                                 CMOVE(3, r0)
PUSH(R0)
                                                                 PUSH (r0)
GETFRAME(-12,R0)
PUSH(R0)
                                                                 DEALLOCATE(1)
                                                                 LD(a, r0)
PUSH (r0)
POP (r0)
POP(R2)
POP(R1)
ADD(R2,R1,R0)
PUSH(R0)
                                                                 WRINT()
POP(R0)
                                                                 DEALLOCATE (0)
ST(R0,a)
                                                                 POP (BP)
                                                                 POP (LP)
POP(LP)
                                                                 RTN()
POP(BP)
DEALLOCATE(2)
RTN()
                                                                 debut:
                                                                 CALL(main)
debut
                                                                 HALT()
                                                                 pile:
                                                                  nom : f; type : void; categorie : fonction; nbparam : 1; nbloc : 2;
CALL(main)
HALT()
                                                                  nom : main; type : void; categorie : fonction;
                                                                  nom : a; type : int; categorie : global; valeur : 10;
                                                                   nom : i; type : int; categorie : param; valeur : ; rang : 0; scope : f;
                                                                  nom : x; type : int; categorie : local; valeur : ; rang : 0; scope : f;
                                                                  nom : y; type : int; categorie : local; valeur : ; rang : 1; scope : f;
Arbre 8:
                                                                 le code généré ne correspond pas.
                                                                  nom : f; type : void; categorie : fonction; nbparam : 2; nbloc : 1;
CMOVE(pile, SP)
                                                                  nom : main; type : void; categorie : fonction;
BR(debut)
                                                                  nom : a; type : int; categorie : global; valeur : 10;
                                                                  nom : x; type : int; categorie : local; valeur : ; rang : 0; scope : f;
|data
                                                                  nom : i; type : int; categorie : param; valeur : ; rang : 0; scope : f;
a: LONG(10)
                                                                  \verb"nom": j; type : int; categorie : param; valeur : ; rang : 1; scope : f;\\
linstructions
```

main: PUSH(LP) PUSH(BP) MOVE(SP,BP) ALLOCATE(0) CMOVE(1, R0) PUSH(R0) CMOVE(2,R0) PUSH(R0) CALL(f) DEALLOCATE(2) POP(R0) ST(R0,a) LD(a,R0) PUSH(R0) PUSH(R0) POP(R0) ST(R0,a) TO(a,R0) PUSH(R0) POP(R0) WRINT() BR(ret_main) ret_main:	
DEALLOCATE(0)	
POP(BP)	
POP(LP)	
RTN()	
f:	
PUSH (LP) PUSH(BP) MOVE(SP,BP) ALLOCATE(1) GETFRAME(-16,R0) PUSH(R0) GETFRAME(-12,R0) PUSH(R0) POP(R2) POP(R1) ADD(R1,R2,R0) PUSH(R0) POP(R0) PUTFRAME(RO, -20) BR(ret_f)	
ret_f:	
DEALLOCATE(1) POP (BP) POP(LP) RTN()	
debut:	
CALL(main) HALT()	

Exemples de programmes sources écrits dans notre langage :

Exemple 1: fonction void main () [] Exemple 2: \$i :: 10; \$j :: 20; \$k; \$I; fonction void main () [] Exemple 3: \$i :: 10; \$j:: 20; \$k; \$I; fonction void main () [\$k::2 \$I::\$i+3.\$j] Exemple 4: \$i; \$j :: 20; fonction void main () [\$i::lire() ecrire(\$i+\$j)] Exemple 5: fonction void main () [\$i::lire() Si(i>10)[ecrire(1) ecrire(2)]] Exemple 6: \$i; \$n :: 5;

```
fonction void main () [
       $i::0
       TantQue($i<$n)[
               ecrire($i)
               $i::$i+1
       ]
]
Exemple 7:
$a::10
fonction void f ($i) [
       $x, $y
       $x::1
       $y::1
       $a::$i+$x+$y
fonction void main () [
       f(3)
       ecrire($a)
]
Exemple 8:
fonction void f ($i, $j) [
       $x
       $x::$i+$j
       retourne($x+10)
fonction void main () [
       $a::f(1,2)
       ecrire($a)
]
```

Bilan critique:

Le projet est inachevé. Cela est dû à une mauvaise gestion du temps et de nombreux problèmes d'organisations et de compréhensions du sujet. Une grande partie de ces problèmes venaient aussi d'une connaissance faible dans le langage assembleur beta et nous avons perdu beaucoup de temps à corriger des erreurs liées à cela.

Conclusion:

De nombreux ralentissements dus à une mauvaise compréhension de certains points du projet nous ont empêché d'aller jusqu'au bout de celui-ci. Malgré cela nous pensons avoir tout de même appris certaines connaissances sur le fonctionnement et la construction d'un compilateur dans un contexte plus général. Enfin, nous avons aussi acquis plus d'expérience en travail de groupe ce qui est aussi un point important pour la suite de nos études. En l'occurrence le début du projet ayant mal démarré nous avons dû faire de nombreux changement par la suite qui nous a pris beaucoup de temps et nous avons dû faire des sacrifices au niveau du temps que nous avons investi dans le projet.