Computer Systems, Spring 2019

Week 5 Lab

Control Structures

Solutions

1 An assignment statement

Here is a small program: just one line of code! The variables have these initial values: x = 0, a = 20, b = 13.

```
x := a - b;
```

Solution.

```
; Computer Systems 1
; Lab Week 5, Control Structures, problem 1

; High level (and low level) program

; Initial values: x = 0, a = 20, b = 13
; x := a-b

; Translation to assembly language

    load   R1,a[R0]    ; R1 := a
    load   R2,b[R0]    ; R2 := b
    sub    R3,R1,R2    ; R3 := a-b
    store  R3,x[R0]    ; x := a-b
    trap   R0,R0,R0    ; halt

x   data    0
a   data   20
b   data   13
```

2 An if-then-else statement

Here is another short program. The variables have these initial values: a = 0, x = 3, y = 20.

```
if x>y
   then { a := x; }
   else { a := y; }
a := 2 * a;
```

Solution.

```
; Computer Systems 1
; Lab Week 5, Control Structures, problem 2

; High level program

; Initial values: a = 0, x = 3, y = 20

; if x > y
;    then { a := x; }
;    else { a := y; }
; a := 2 * a

; Translation to low level program

; Initial values: a = 0, x = 3, y = 20

;    b := x>y
;    if b=False then goto ElsePart
;    a := x
;    goto AfterIf
; ElsePart
;    a := y
; AfterIf
;    a := 2 * a

; Translation to assembly language

; Register usage
; R1 = x
; R2 = y
; R3 = a
; R4 = temporary

; b := x>y
     load   R1,x[R0]          ; R1 := x
     load   R2,y[R0]          ; R2 := y
     cmpgt  R4,R1,R2          ; R4 = b := (x>y)

; if b=False then goto ElsePart
     jumpf  R4,ElsePart[R0]   ; if b=False then goto ElsePart

; a := x
     load   R3,x[R0]          ; R3 := x
     store  R3,a[R0]          ; a := x

; goto AfterIf
     jump   AfterIf[R0]       ; goto AfterIf

ElsePart
```

```
; a := y
     load   R3,y[R0]          ; R3 := y
     store  R3,a[R0]          ; a := y

AfterIf

; a := 2 * a
    lea    R4,2[R0]           ; R4 := 2
    mul    R4,R4,R3           ; R4 := 2 * a
    store  R4,a[R0]           ; a := 2 * a

    trap   R0,R0,R0           ; halt

x   data    3
y   data   20
a   data    0
```

3 A while loop

Here is a program that contains a while loop. The initial value of `n` is 5; the other variables have initial value 0.

```
sum := 0;
i := 0;
while i < n do
   { sum := sum + i;
     i := i + 1;
   }
```

Solution.

```
; Computer Systems 1
; Lab Week 5, Control Structures, problem 3

; High level program

; initial: n = 5

; sum := 0
; i := 0
; while i < n do
;    { sum := sum + i;
;      i := i + 1;
;    }

; Translation to low level program

; initial: n = 5
```

```
;    sum := 0;
;    i := 0;
; loop
;    if not (i<n) then goto done
;    sum := sum + i;
;    i := i + 1;
;    goto loop
; done

; Translation to assembly language

; Register usage
;    R1 = sum
;    R2 = i
;    R3 = n
;    R4 = 1

     add    R1,R0,R0      ; sum := 0
     add    R2,R0,R0      ; i := 0
     load   R3,n[R0]      ; R3 := n
     lea    R4,1[R0]      ; R4 := 1

loop
; if not (i<n) then goto done
     cmplt  R4,R2,R3      ; R4 := (i<n)
     jumpf  R4,done[R0]   ; if (i<n)=False then goto done

     add    R1,R1,R2      ; sum := sum + i
     add    R2,R2,R4      ; i := i + 1
     jump   loop[R0]      ; goto loop
done
     store  R1,sum[R0]    ; sum := R1
     trap   R0,R0,R0      ; halt

sum  data   0
n    data   5
```