02321 Hardware/Software Programmering (E10)

Home assignment 8

Morten Hillebo (s072923)
Kim Rostgaard Christensen (s084283)

Group 2

Problem 14.2

(a)

The dynamic link is a copy of the frame pointer of the calling function. This copy is needed for the caller function to be able to access its own variables after end of function call.

(b)

The purpose of the return address is to provide a way back to the caller function, even if the callee calls another function.

The return address is pushed on the stack by the caller. When the callee finishes, it pops the return address off the stack . Puts it into R7, and does RET.

(c)

a) The return value is modified by the callee and saved in the Activation Record of the caller. The purpose of the return value is to give back a result from the callee to the caller.

Problem 14.4 - caller or callee action

(a) Writing arguments into the activation record

Caller

(b) Writing the return value

Callee

(c) Writing the dynamic link

Callee

(d) Modifying the value in R5 to point within the called function's activation record

Callee

Problem 14.8

The code outputs:

```
x = 1   y = 2
```

Which is the unaltered values. This is due to the fact that the function Swap only swaps the values in the parameter scope. When the function returns the paramter values x and y of Swap is abandoned. This can be solved by using pointers instead.

Problem 14.10

(a)

```
ADD R6, R6, #–2 ; Allocate space for the return value
STR R7, R6, #0  ; Push the return address
ADD R6, R6, #–1 ; Allocate space for the dynamic link
STR R5, R6, #0  ; Push the dynamic link
ADD R5, R6, #–1 ; Set new frame pointer
ADD R6, R6, #–4 ; This is allocation for local variables
```

Short answer; 4.

(b)

```
LDR R0, R5, #4  ; Load x
LDR R1, R5, #5  ; Load y
ADD R0, R0, R1  ; Add them, and store in R0 (x + y)
```

Problem 14.14

See attached code.

Problem 15.6

```
a:75, b:60, c:72
a:57, b:46, c:50
a:55, b:45, c:97
a:53, b:43, c:55
a:44, b:36, c:44
a:37, b:31, c:63
a:36, b:30, c:48
a:35, b:30, c:92
a:33, b:28, c:62
a:24, b:21, c:45
a:21, b:20, c:90
a:17, b:16, c:46
a:16, b:15, c:37
```