605.204 - Computer Organization
Module 8: Assignment

Nick Hinke

October 23, 2022

Brief Introduction

This assignment involves the definition of global symbols for use across various assembly files in armv7l. All of my resulting code can be found at `https://github.com/nhinke/computer-organization-repo/tree/master/assignments/module08` and can be cloned (along with pre-built binaries in a *bin/* folder) and viewed using the following commands:

```
git clone https://github.com/nhinke/computer-organization-repo.git
cd computer-organization-repo/assignments/module08/
```

The pre-built binaries can then be run using the following commands:

```
cd bin/
./kphTest
./CToFTest
./InchesToFeetTest
```

Note that each of the pre-built binaries will print out an example input-output sequence to the active terminal.

Problem 1

1. The first function is miles2kilometer(int miles), which will convert kilometers to miles by multiplying by 100 and dividing by 166. (Put in a comment why we multiply by 100, then divide by 166. Suggest how even greater precision could be achieved using this method for integer numbers). The second function is kph(int hours, int miles), and must be calculated by calling function miles2kilometers to convert the miles to kilometers, and then dividing by hours. Write a main program to prompt for and read the values of miles and hours, and print out the mph.

miles2kilometers

Note that the number of kilometers is equivalent to the number of miles multiplied by 1.6029. However, since we are dealing strictly with integers, we cannot accomplish this conversion directly. Rather, we could first multiply by 16 and subsequently divide by 10 (which is equivalent to multiplying by 1.6), or we could achieve greater precision by first multiplying by 161 and then dividing by 100. In fact, we could retain greater accuracy still by multiplying by 1609 first and then dividing by 1000.

Functions in library program:

```
1  # Nick Hinke
2  # 10/23/2022
3  # 605.204 Computer Organization
4  # Module 8 Assignment
5  #
6  # Library of conversion functions used in each problem
7  #
8
9  .global miles2kilometers
10 .global kph
11 .global CToF
12 .global InchesToFeet
13
14
15 .text
16 # function to convert miles (in r0) to kilometers (returned in r0)
17 miles2kilometers:
18
19   # Note that: numKM = numMI*1.60934
20   # However, since we are strictly dealing with integers, we cannot multiply by 1.60934 directly.
21   # We could multiply by 16 and subsequently divide by 10 (equivalent to multiplying by 1.6).
22   # However, we could retain more accuracy by multiplying by 161 and dividing by 100 (equivalent to 1.61).
23   # That being said, we could do better still by multiplying by 1609 and dividing by 1000 (equivalent to 1.609).
24
25   # push stack
26   SUB sp, sp, #4
27   STR lr, [sp, #0]
28
29   # convert mi to km via: r0 = r0*161/100
30   MOV r1, #161
31   MUL r0, r0, r1     @ r0 = miles*161
32   MOV r1, #100
33   BL __aeabi_idiv    @ r0 = miles*161/100
34
35   # pop stack and return
36   LDR lr, [sp, #0]
37   ADD sp, sp, #4
38   MOV pc, lr
39
40 .data
41 # end miles2kilometers
```

Figure 1: Screenshot of miles to kilometers program

```
44  .text
45  # function to convert hours (in r0) and miles (in r1) to kph (returned in r0)
46  kph:
47
48    # push stack
49    SUB sp, sp, #4
50    STR lr, [sp, #0]
51
52    # convert hours and miles to kph via: r0 = miles2kilometers(r1)/r0
53    LDR r2, =numHours
54    STR r0, [r2, #0]     @ store hours in numHours
55    MOV r0, r1           @ move miles to r0
56    BL miles2kilometers  @ now kilometers in r0
57    LDR r1, =numHours
58    LDR r1, [r1, #0]     @ move hours to r1
59    BL __aeabi_idiv      @ r0 = miles2kilometers(miles)/hours
60
61    # pop stack and return
62    LDR lr, [sp, #0]
63    ADD sp, sp, #4
64    MOV pc, lr
65
66  .data
67    numHours: .word 0
68  # end kph
```

Figure 2: Screenshot of miles and hours to kph program

Example:

```
.text
.global main
main:

  # push stack
  SUB sp, sp, #4
  STR lr, [sp, #0]

  # print number of hours prompt
  LDR r0, =promptHr
  BL printf

  # read in number of hours
  LDR r0, =formatHr
  LDR r1, =numHr
  BL scanf

  # print number of miles prompt
  LDR r0, =promptMi
  BL printf

  # read in number of miles
  LDR r0, =formatMi
  LDR r1, =numMi
  BL scanf

  # load number of hours into r0 and number of miles into r1
  LDR r0, =numHr
  LDR r1, =numMi
  LDR r0, [r0, #0]
  LDR r1, [r1, #0]

  # call kph function in libConversions and print results
  BL kph
  MOV r1, r0
  LDR r2, =numHr
  LDR r3, =numMi
  LDR r2, [r2, #0]
  LDR r3, [r3, #0]
  LDR r0, =outputKph
  BL printf

  # pop stack and return
  LDR lr, [sp, #0]
  ADD sp, sp, #4
  MOV pc, lr

.data
  outputKph: .asciz "You will need to travel at %d kph for %d hours to travel %d miles\n"
  promptHr: .asciz "Please enter the number of hours you will be traveling: "
  promptMi: .asciz "Please enter the number of miles you would like to go:  "
  formatHr: .asciz "%d"
  formatMi: .asciz "%d"
  numHr: .word 0
  numMi: .word 0
# end main
```

Figure 3: kph test program

```
rpi@rpi1:~/Documents/JHU/Computer-Organization/computer-organization-repo/assignments/module08/bin $ ./kphTest
Please enter the number of hours you will be traveling: 3
Please enter the number of miles you would like to go:  35
You will need to travel at 18 kph for 3 hours to travel 35 miles
```

Figure 4: kph test output

Problem 2

2. Write the functions CToF and InchesToFeet and add it to the conversions.s file. Write a main program to call it and test it.

Functions in library program:

```
71  .text
72  # function to convert degrees Celsius (in r0) to degrees Fahrenheit (returned in r0)
73  CToF:
74
75    # push stack
76    SUB sp, sp, #4
77    STR lr, [sp, #0]
78
79    # convert Celsius to Fahrenheit via: r0 = r0*9/5+32
80    MOV r1, #9
81    MUL r0, r0, r1      @ r0 = Celsius*9
82    MOV r1, #5
83    BL __aeabi_idiv     @ r0 = Celsius*9/5
84    ADD r0, r0, #32     @ r0 = Celsius*9/5+32
85
86    # pop stack and return
87    LDR lr, [sp, #0]
88    ADD sp, sp, #4
89    MOV pc, lr
90
91  .data
92  #end CToF
```

Figure 5: Screenshot of CToF program

```
 95  .text
 96  # function to convert inches (in r0) to feet and inches (returned in r0 and r1, respectively)
 97  InchesToFeet:
 98
 99    # push stack
100    SUB sp, sp, #4
101    STR lr, [sp, #0]
102
103    # get number of feet via: r0 = r0/12
104    LDR r2, =numInches
105    STR r0, [r2, #0]    @ store inches in numInches
106    MOV r1, #12
107    BL __aeabi_idiv     @ r0 = inches/12 = feet
108
109    # get number of inches remaining via: r1 = inches-(r0*12)
110    MOV r1, #12
111    MUL r3, r0, r1      @ r3 = feet*12
112    LDR r2, =numInches
113    LDR r2, [r2, #0]    @ r2 = inches
114    SUB r1, r2, r3      @ r1 = inches-(feet*12) = inches remaining
115
116    # pop stack and return
117    LDR lr, [sp, #0]
118    ADD sp, sp, #4
119    MOV pc, lr
120
121  .data
122    numInches: .word 0
123  #end InchesToFeet
```

Figure 6: Screenshot of InchesToFeet program

Examples:

```
# Nick Hinke
# 10/23/2022
# 605.204 Computer Organization
# Module 8 Assignment - Problem 2a
#
# Test library function (CToF) written to convert Celsius to Fahrenheit
#

.text
.global main
main:

  # push stack
  SUB sp, sp, #4
  STR lr, [sp, #0]

  # print degrees Celsius prompt
  LDR r0, =promptCels
  BL printf

  # read in degrees Celsius
  LDR r0, =formatCels
  LDR r1, =degCels
  BL scanf

  # load degrees Celsius into r0
  LDR r0, =degCels
  LDR r0, [r0, #0]

  # call CToF function in libConversions and print results
  BL CToF
  MOV r2, r0
  LDR r1, =degCels
  LDR r1, [r1, #0]
  LDR r0, =outputFahr
  BL printf

  # pop stack and return
  LDR lr, [sp, #0]
  ADD sp, sp, #4
  MOV pc, lr

.data
  outputFahr: .asciz "%d degrees Celsius is equivalent to: %d degrees Fahrenheit\n"
  promptCels: .asciz "Please enter temperature in degrees Celsius: "
  formatCels: .asciz "%d"
  degCels: .word 0
# end main
```

Figure 7: CToF test program

```
rpi@rpi1:~/Documents/JHU/Computer-Organization/computer-organization-repo/assignments/module08/bin $ ./CToFTest
Please enter temperature in degrees Celsius: 100
100 degrees Celsius is equivalent to: 212 degrees Fahrenheit
```

Figure 8: CToF test output

```
# Nick Hinke
# 10/23/2022
# 605.204 Computer Organization
# Module 8 Assignment - Problem 2b
#
# Test library function (InchesToFeet) written to convert inches to feet and inches
#

.text
.global main
main:

  # push stack
  SUB sp, sp, #4
  STR lr, [sp, #0]

  # print number of inches prompt
  LDR r0, =promptIn
  BL printf

  # read in number of inches
  LDR r0, =formatIn
  LDR r1, =numIn
  BL scanf

  # load number of inches into r0
  LDR r0, =numIn
  LDR r0, [r0, #0]

  # call InchesToFeet function in libConversions and print results
  BL InchesToFeet
  MOV r2, r0
  MOV r3, r1
  LDR r1, =numIn
  LDR r1, [r1, #0]
  LDR r0, =outputFtIn
  BL printf

  # pop stack and return
  LDR lr, [sp, #0]
  ADD sp, sp, #4
  MOV pc, lr

.data
  outputFtIn: .asciz "%d inches is equivalent to: %d feet and %d inches\n"
  promptIn: .asciz "Please enter total number of inches: "
  formatIn: .asciz "%d"
  numIn: .word 0
# end main
```

Figure 9: InchesToFeet test program

```
rpi@rpi1:~/Documents/JHU/Computer-Organization/computer-organization-repo/assignments/module08/bin $ ./InchesToFeetTest
Please enter total number of inches: 29
29 inches is equivalent to: 2 feet and 5 inches
```

Figure 10: InchesToFeet test output