605.204 - Computer Organization
Module 10: Assignment

Nick Hinke

November 05, 2022

Brief Introduction

This assignment involves branching and looping within various assembly files in armv7l. All of my resulting code can be found at this *GitHub link* 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/module10/
```

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

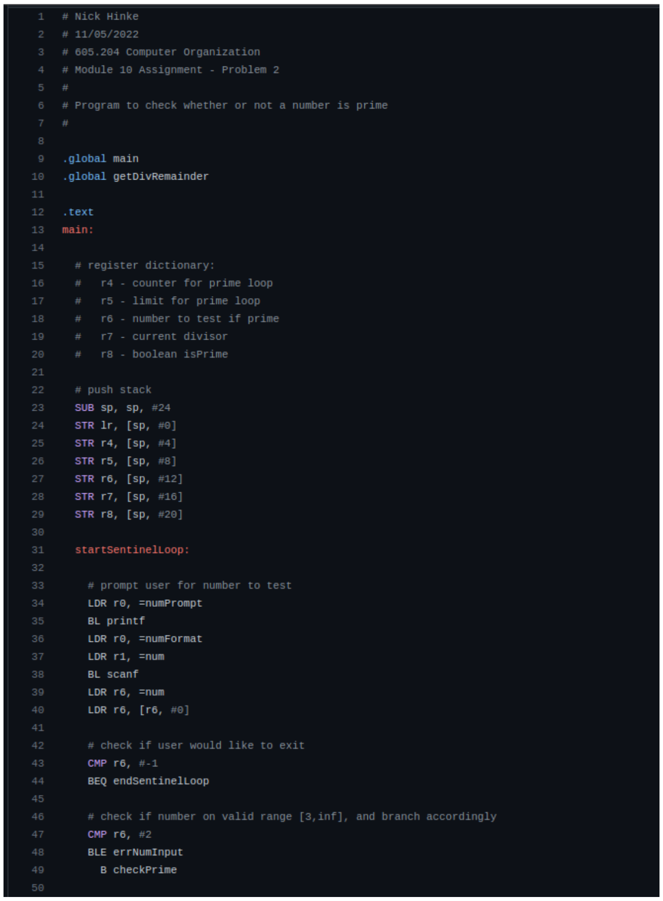
```
cd bin/
./pr2-checkPrime
./pr3-guessNumber
```

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

Problem 2

2. Write a program to prompt the user for a number, and determine if that number is prime. Your program should print out "Number n is prime" if the number is prime, and "Number n is not prime" if the number is not prime. The user should be able to enter a "-1" to end the problem. It should print an error if 0, 1, 2 or any negative number other than -1 are entered.

Top of Program:

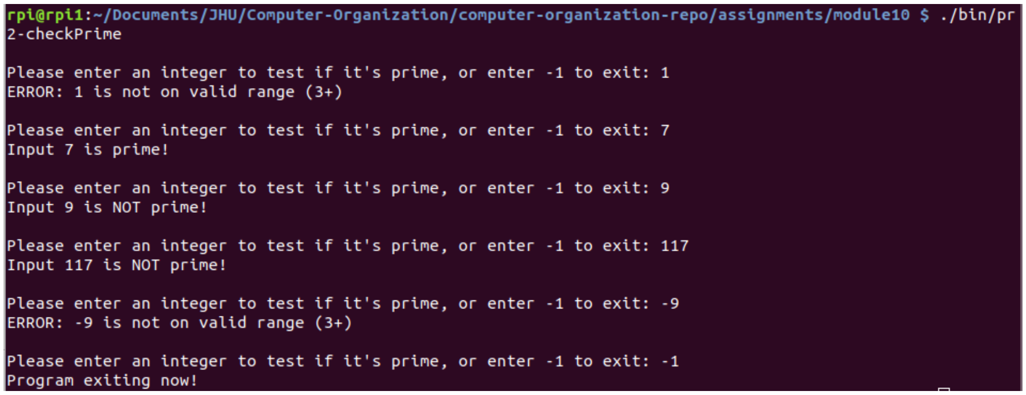


```
# Nick Hinke
# 11/05/2022
# 605.204 Computer Organization
# Module 10 Assignment - Problem 2
#
# Program to check whether or not a number is prime
#

.global main
.global getDivRemainder

.text
main:

  # register dictionary:
  #   r4 - counter for prime loop
  #   r5 - limit for prime loop
  #   r6 - number to test if prime
  #   r7 - current divisor
  #   r8 - boolean isPrime

  # push stack
  SUB sp, sp, #24
  STR lr, [sp, #0]
  STR r4, [sp, #4]
  STR r5, [sp, #8]
  STR r6, [sp, #12]
  STR r7, [sp, #16]
  STR r8, [sp, #20]

  startSentinelLoop:

    # prompt user for number to test
    LDR r0, =numPrompt
    BL printf
    LDR r0, =numFormat
    LDR r1, =num
    BL scanf
    LDR r6, =num
    LDR r6, [r6, #0]

    # check if user would like to exit
    CMP r6, #-1
    BEQ endSentinelLoop

    # check if number on valid range [3,inf], and branch accordingly
    CMP r6, #2
    BLE errNumInput
      B checkPrime

```

Figure 1: Screenshot of program to check if number is prime

Visit *this link* to see the rest of the program (it is far too long to paste screenshots in a meaningful way).

Example:



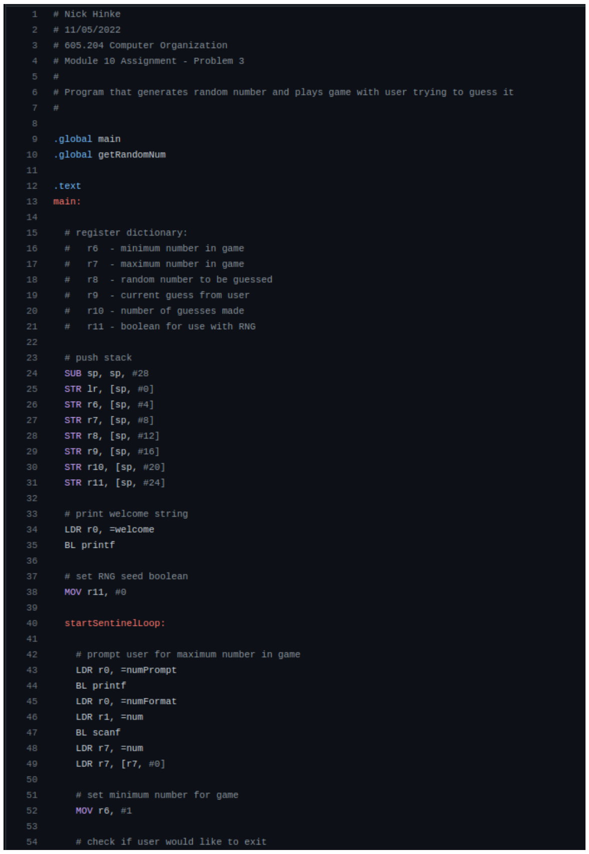
```
rpi@rpi1:~/Documents/JHU/Computer-Organization/computer-organization-repo/assignments/module10 $ ./bin/pr
2-checkPrime

Please enter an integer to test if it's prime, or enter -1 to exit: 1
ERROR: 1 is not on valid range (3+)

Please enter an integer to test if it's prime, or enter -1 to exit: 7
Input 7 is prime!

Please enter an integer to test if it's prime, or enter -1 to exit: 9
Input 9 is NOT prime!

Please enter an integer to test if it's prime, or enter -1 to exit: 117
Input 117 is NOT prime!

Please enter an integer to test if it's prime, or enter -1 to exit: -9
ERROR: -9 is not on valid range (3+)

Please enter an integer to test if it's prime, or enter -1 to exit: -1
Program exiting now!
```

Figure 2: Screenshot of program output

Problem 3

3. Write a program to allow a user to guess a random number generated by the computer from 1 to "maximum" (the user should enter the maximum value to guess). In this program the user will enter the value of maximum. The user will then enter guesses and the program should print out if the guess is too high or too low until the user guesses the correct number. The program should print out the number of guesses the user took.

Top of Program:

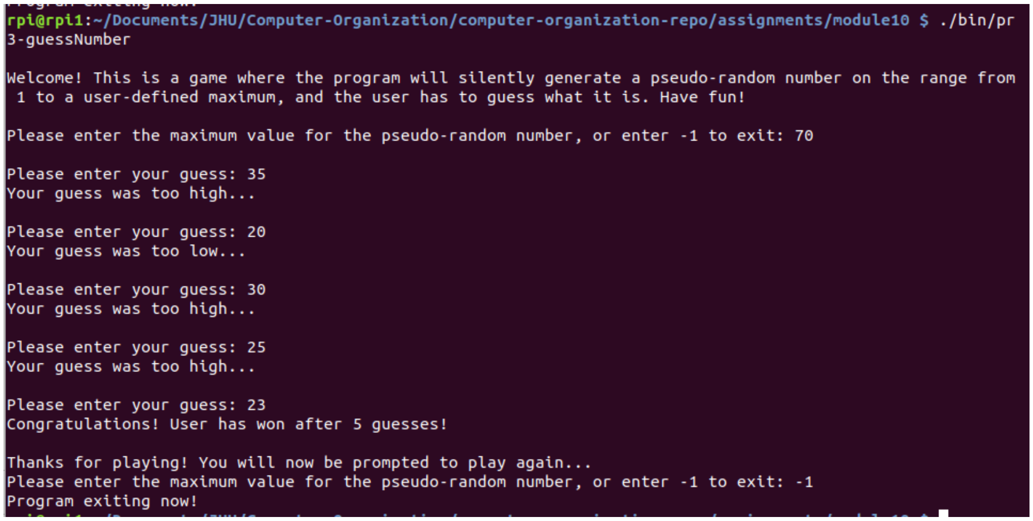


```
# Nick Hinke
# 11/05/2022
# 605.204 Computer Organization
# Module 10 Assignment - Problem 3
#
# Program that generates random number and plays game with user trying to guess it
#

.global main
.global getRandomNum

.text
main:

  # register dictionary:
  #   r6  - minimum number in game
  #   r7  - maximum number in game
  #   r8  - random number to be guessed
  #   r9  - current guess from user
  #   r10 - number of guesses made
  #   r11 - boolean for use with RNG

  # push stack
  SUB sp, sp, #28
  STR lr, [sp, #0]
  STR r6, [sp, #4]
  STR r7, [sp, #8]
  STR r8, [sp, #12]
  STR r9, [sp, #16]
  STR r10, [sp, #20]
  STR r11, [sp, #24]

  # print welcome string
  LDR r0, =welcome
  BL printf

  # set RNG seed boolean
  MOV r11, #0

  startSentinelLoop:

    # prompt user for maximum number in game
    LDR r0, =numPrompt
    BL printf
    LDR r0, =numFormat
    LDR r1, =num
    BL scanf
    LDR r7, =num
    LDR r7, [r7, #0]

    # set minimum number for game
    MOV r6, #1

    # check if user would like to exit
```

Figure 3: Screenshot of program to play number guessing game

Visit *this link* to see the rest of the program (it is far too long to paste screenshots in a meaningful way).

Example:



```
rpi@rpi1:~/Documents/JHU/Computer-Organization/computer-organization-repo/assignments/module10 $ ./bin/pr
3-guessNumber

Welcome! This is a game where the program will silently generate a pseudo-random number on the range from
 1 to a user-defined maximum, and the user has to guess what it is. Have fun!

Please enter the maximum value for the pseudo-random number, or enter -1 to exit: 70

Please enter your guess: 35
Your guess was too high...

Please enter your guess: 20
Your guess was too low...

Please enter your guess: 30
Your guess was too high...

Please enter your guess: 25
Your guess was too high...

Please enter your guess: 23
Congratulations! User has won after 5 guesses!

Thanks for playing! You will now be prompted to play again...
Please enter the maximum value for the pseudo-random number, or enter -1 to exit: -1
Program exiting now!
```

Figure 4: Screenshot of sample program output