Name: Cameron Peterson-Zopf

PROGRAMMING ASSIGNMENT 3

Discussion on a high level with your colleagues is encouraged. Make sure the work submitted is your own. When in doubt, ask a TA or the instructor. If you are not sure what constitutes academic dishonesty, please refer to the AISC web site: https://aisc.uci.edu/.

You can fill out your answers below in text, paste screenshots, and/or include images (make sure the image is right side up & legible).

This homework covers:

LC-3 Machine Language to Assembly – Part 1

AISC

Please initial here to indicate you understand UCI's Academic Integrity Policy and confirm that this is your own work you are submitting (this counts for points): CPZ

UPDATED CODE SCREENSHOT

Final assembly code w/ comments (screenshot or pasted in)

```
hello-buddy-login.asm
1 ;Problem 9.30
 3
    ; Program asks user to enter their name, then greets the user:
 4
    ; > Please enter your name: <user input>
       > Hello, <user input>
    ; > Hello, <user input> ; Encription: encripted by having the ASCII value of the character be stored as #10 more.
 8
    ; R0: holds the input (from TRAP x20) & uses as output (for TRAP x21) and starting addresses for output strings
    ; R1: initializes to the starting of HELLO string & ends up pointing to the end of the string. Will be saved in memory.
    ; R2: temporary register used for intermediate calculations. Also used as a counter for attempts. Will be saved in memory.
 10
    ; R3: temporary register used for intermediate calculations. Also used to store info to jump to SUCCESS. Will be saved in memory.
    ; R4: temporary register used for intermediate calculations. Also used to store info to jump to YESPW
    ; R5: temporary register used for intermediate calculations
    R6: temporary register used for intermediate calculations; R7: reserved for subroutines
 15
16
             .ORIG
                     x3000
                     R1, HELLO
18
             LEA
                                     ; starting address of string HELLO (going to find end of string for appending)
19
             AND
                     R2, R2, x0000
                                     ; R2<-0
                                     ; R2<-3
             ADD
20
                     R2, R2, #3
                                     ; save R2 as it will be used in subroutines for user
21
             ST
                     R2, Save
                     R2, SaveP
                                     ; save R2 as it will be used in subroutines for password
                     R3, SUCCESS
 23
             LEA
                                   ; save where SUCCESS is
                                    ; save R3 as it will be used in subroutines
                     R3, Save1
                                     ; save where YESPW is
25
             LEA
                    R4, YESPW
R4, Save2
                                     ; save R4 as it will be used in subroutines
26
```

EECS 20 Homework Page 1 of 12

```
; get to end of HELLO string for writing/appending input
27
                                   ; load char at hello address into R2
28 AGATN
          I DR
                   R2,R1,#0
29
           BRz
                   NEXT
                                    ; if done w/ string (x0000...NULL), go to NEXT
                                    ; increment hello address..so you can store next character into R2, overwriting previous
30
            ΔDD
                   R1,R1,#1
31
           ST
                   R1, SaveR1
                                    ; save value of R1 to point to starting location for user input.
32
           BR
                   AGAIN
33
           ; when we branch to NEXT, R1 will continue the address of the NULL character in the hello string.
34
35
    ; print prompt for username input
36
            LEA
                   R0, PROMPT
                                   ; get starting address of prompt for user
   NEXT
37
           TRAP
                    x22
                                    ; PUTS (output entire welcome message)
38
39
40
   ;Get user input and compare
41
                                    ; call reading in of username
                                    ; compare against USER1
42
            JSR
                    FuCom1
                                    ; compare against USER2
43
            JSR
                    FuCom2
44
            JSR
                    FuCom3
                                    ; compare against USER3
   ; If not a valid username...
45
46
            LD
                    R2, Save
                                    ; load back R2.
                                    ; decrement R2
47
            ADD
                    R2, R2, #-1
                                    ; tried 3 times
48
           BRz
                    FINISH
                    R2, Save
                                    ; save R2 again
49
           ST
                   R0,FAIL
                                    ; get starting address of fail message for user
50
           LEA
51
            TRAP
                   x22
                                   ; PUTS (output)
52
            JSR
                   ClrU
                                    ; clear previous username
53
            BR
                   NEXT
                                     back to start
54
   FINISH
           LEA
                   R0,FAIL3
                                    ; get starting address of fail 3 times message for user
55
            TRAP
                   x22
                                     PUTS (output)
56
           BR
                   BYF
57
    ; Get password input and compare
58
   SUCCESS LEA
                   R0, PWPROMPT
59
            TRAP
                   x22
60
            JSR
                   FuPW
                                   ; call reading in of password
                                   ; compare against PW1
61
           ISR
                   PWCom1
                                    ; compare against PW2
62
           JSR
                   PWCom2
63
           ISR
                   PWCom3
                                    ; compare against PW3
64
   ; If not a valid password...
                   R2, SaveP
65
           LD
                                   ; load back R2.
            ADD
                   R2, R2, #-1
66
                                     decrement R2
67
           BR7
                   FTNTSH1
                                     tried 3 times
68
            ST
                   R2, SaveP
                                     save R2 again
                                     get starting address of fail message for user
69
            LEA
                   R0.FAIL
            TRAP
70
                                     PUTS (output)
                   x22
                   ClrP
                                    ; clear previous password
71
            JSR
            BR
                   SUCCESS
                                     back to start
72
                                    ; get starting address of fail 3 times message for user
   FINISH1 LEA
73
                   R0.FAIL3
74
            TRAP
                   x22
                                    ; PUTS (output)
75
           BR
                   BYE
76
    : password was correct
                                   ; address of HELLO Prompt
77
                   RØ, HELLO
x22
   YESPW
           LEA
           TRAP
                                   ; PUTS...since we overwrote the 1st NULL, this will print out both Hello, name
78
79
           LEA
                   RØ, LOGIN
                                   ; address of LOGIN Prompt
80
            TRAP
                   x22
   BYE
81
                   x25
                                   ; HALT (stop program)
82
83
84
          85
    ;function for writing username
86
                   R3, NEGENTER
   FuUS
                                    store (through ld) NEGENTER into R3 for termination comparing later
87
           LD
                                    load in starting address for input.
88
            LD
                   R1. SaveR1
89
   AGAIN2
           TRAP
                                    GETC (gather user input one char at a time)
                   x20
                                    OUT (output char for user on console)
90
            TRAP
                   x21
91
            ADD
                   R2,R0,R3
                                    check if user pressed ENTER key. If R2 = 0 => R0 = x0A
92
                   ADDNULL
93
           STR
                   R0,R1,#0
                                    store value in R0 into memory. R1 is first pointing to NULL character, which will be overwritten.
94
            ADD
                   R1,R1,#1
                                  ; increment address of R1 so that we can write to the next available spot (in .BLKW #25 area)
95
           RR
                   AGAIN2
                   R6, R6, x0000
96
   ADDNULL
           AND
                                  ; R6<-0
                                   ; end with NULL
97
           STR
                   R6, R1, #0
98
           RET
99
```

EECS 20 Homework Page 2 of 12

```
100 ;function for writing password
                                   ; store (through ld) NEGENTER into R3 for termination comparing later
                    R3, NEGENTER
101 FuPW
            LD
                    R5, SPACE
             LEA
                                     store in empty space. Load back in starting location.
 102
                    x20
             TRAP
                                     GETC (gather user input one char at a time)
 103
     AGAIN3
                    R2,R0,R3
 104
             ADD
                                     check if user pressed ENTER key. If R2 = 0 => R0 = x0A
 105
             BRz
                    NULLPW
                                    ; if they did, we are done
 106
             STR
                    R0,R5,#0
                                    ; store value in R0 (user input) into memory.
 107
             ADD
                    R5,R5,#1
                                    ; increment address of R5 so that we can write to the next available spot
 102
             RR
                    AGAIN3
 109 NULLPW
                    R6, R6, x0000
            AND
                                   ; R6<-0
                                   ; end with NULL
110
             STR
                    R6, R5, #0
 111
             RET
 112
 113
     ;******************Subroutines for Clearing Wrong Username and PW**************
     ;function for clearing username
                    R3, NEGENTER
                                   ; store (through ld) NEGENTER into R3 for termination comparing later
 115
     ClrU
            LD
 116
             ΙD
                    R1, SaveR1
                                     load in starting address for input.
 117
             AND
                    R2,R2, x0000
                                     R2<-0
 118
             ADD
                    R4,R2, #15
                                     R4<-15
                                   ; R4<-25
             ADD
                    R4, R4, #10
 119
 120 AGAIN4
             STR
                    R2,R1,#0
                                     Set value to zero.
 121
             ADD
                    R1,R1,#1
                                    ; increment address of R1 so that we can write to the next available spot (in .BLKW #25 area)
 122
             ADD
                    R4, R4, #-1
                                    ;decrement R4 counter
123
             BRz
                                   ; done clearing all 25 spaces.
 124
             BR
                    AGAIN4
 125
             RET
126
127
128 ;function for clearing password
129 ClrP
             LD
                    R3, NEGENTER
                                   ; store (through ld) NEGENTER into R3 for termination comparing later
                     R5, SPACE
                                     ; load in starting address for input pw
                                    ; R2<-0
131
             AND
                     R2, R2, x0000
132
             ADD
                     R4,R2, #15
                                    ; R4<-25
133
             ADD
                     R4,R4, #10
134 AGAIN5
             STR
                     R2, R5, #0
                                    ; Set value to zero.
135
             ADD
                     R5, R5, #1
                                     ; increment address of R1 so that we can write to the next available spot (in .BLKW #25 area)
             ADD
                                    ;decrement R4 counter
136
                     R4, R4, #-1
137
             BRz
                                    ; done clearing all 25 spaces.
138
             BR
                    AGAIN5
139
             RET
140
     141
     ; function for comparing input to USER1
142
143
     FuCom1 LEA
                    R2, USER1
                                    ; load starting address of USER1 into R2
144
             LEA
                    R3, INPUT
                                    ; load address of 2nd character of input (1st character overwrote NULL)
                     R3, R3, #-1
145
             ADD
                                    ; Now we have starting address of input
146 Loop1
             LDR
                     R4, R2, #0
                                    ; R4<-M[R2] // R4<-M[address]...store character from USER1 into R4
147
             ADD
                     R2, R2, #1
                                    ; increments R2 to next address
148
             LDR
                     R5, R3, #0
                                    ; R5<-M[R3]...store character from input into R5
149
             ADD
                     R3, R3, #1
                                    ; increments R3 to next address
150
             NOT
                     R5, R5
151
             ΔDD
                     R5, R5, #1
                                    ; 2s comp => negative
152
             ADD
                    R6, R4, R5 ; If R6 = 0 => same character
 153
               BRnp
                       NOTUSER1
                                       ; not a valid username
                       R4, R4, #0
                                       ; List R4 again for branch statement
 154
              ADD
              BRz
 155
                       SUCCESS
                                        ; if R4 and R5 is the NULL character, input is done, and matched with USER1
 156
              RR
                       Loop1
 157 NOTUSER1 RET
                                        ; the input was not this username, try to compare against the next one.
 158
 159
       ; function for comparing input to USER2
                                       ; load starting address of USER2 into R2
 160
     FuCom2 LEA
                       R2, USER2
                                        ; load address of 2nd character of input (1st character overwrote NULL)
 161
              LEA
                       R3, INPUT
 162
               ADD
                       R3, R3, #-1
                                         Now we have starting address of input
 163
     Loop2
              LDR
                       R4, R2, #0
                                         R4<-M[R2] // R4<-M[address]...store character from USER2 into R4
 164
               ADD
                       R2, R2, #1
                                        ; increments R2 to next address
               LDR
                       R5, R3, #0
                                       ; R5<-M[R3]...store character from input into R5
 165
 166
               ADD
                       R3, R3, #1
                                       ; increments R3 to next address
                       R5 R5
 167
              NOT
              ADD
 168
                       R5, R5, #1
                                        ; 2s comp => negative
               ADD
 169
                       R6, R4, R5
                                       ; If R6 = 0 => same character
 170
               BRnp
                       NOTUSER2
                                        ; not a valid username
                       R4, R4, #0
 171
               ADD
                                        ; List R4 again for branch statement
                       SUCJMP1
 172
               BRz
                                        ; if R5 is the NULL character, input is done, and matched with USER2
 173
              BR
                       Loop2
 174 NOTUSER2 RET
                                        ; the input was not this username, try to compare against the next one.  \\
 175 SUCJMP1 LD
                       R3, Save1
                                        ; load back R3, which is address for SUCCESS.
              ПМР
 176
                       R3
                                         ; jump to where success is.
 177
```

EECS 20 Homework Page 3 of 12

```
178 ; function for comparing input to USER3
                                 ; load starting address of USER3 into R2
 179 FuCom3 LEA
                     R2, USER3
                     R3, INPUT
180
             LEA
                                    ; load address of 2nd character of input (1st character overwrote NULL)
                                   ; Now we have starting address of input
             ADD
                     R3, R3, #-1
 181
 182 Loop3
             LDR
                     R4, R2, #0
                                    ; R4<-M[R2] // R4<-M[address]...store character from USER3 into R4
183
             ADD
                     R2, R2, #1
                                    ; increments R2 to next address
             LDR
                     R5, R3, #0
                                    ; R5<-M[R3]...store character from input into R5
 184
 185
             ADD
                     R3, R3, #1
                                    ; increments R3 to next address
                     R5 R5
 186
             NOT
             ADD
                     R5, R5, #1
                                    ; 2s comp => negative
; If R6 = 0 => same character
 187
                     R6, R4, R5
188
             ADD
                                     ; not a valid username
 189
             BRnp
                     NOTUSER3
 190
             ADD
                     R4, R4, #0
                                    ; List R4 again for branch statement
                     SUCCIMP
 191
             BRz
                                     ; if R5 is the NULL character, input is done, and matched with USER3
 192
             BR
                     Loop3
 193 NOTUSER3 RET
                                     ; the input was not this username, try to compare against the next one.
                                     ; load back R3, which is address for SUCCESS.
 194 SUCCJMP LD
                     R3, Save1
 195
        JMP
                     R3
                                      ; jump to where success is.
196
197 ;************memory storage****************************
 198 NEGENTER
                 .FILL xFFF6
                                     ; -x0A...negative of ENTER key
 199 Save
                 .BLKW
                            1
 200 SaveP
                 .BLKW
 201 Save1
                 .BLKW
 202 Save2
                 .BLKW
                            1
                 .BLKW
 203 SaveR1
 204 SPACE
                 .BLKW
                            #25
 205 HELLO
                .STRINGZ
                            "Hello, "
 206 INPUT
                 .BLKW
                            #25
                            "Enter your username:
 207 PROMPT
                 .STRINGZ
                            "Not valid, try again "
 208 FAIL
                 .STRINGZ
 209 FAIL3
                 .STRINGZ
                            "3 attempts failed, goodbye"
                            "Enter your password: '
 210 PWPROMPT
                .STRINGZ
 211 LOGIN
                 .STRINGZ
                            ". You have logged in"
                            "panteater"
 212 USER1
                 .STRINGZ
                            "ujyjw"
 213 PW1
                .STRINGZ
                                            ; "peter"
 214 USER2
                 .STRINGZ
                            "qv"
 215 PW2
                .STRINGZ
                            "mjqqtymjwj&"
                                           ; "hellothere!"
 216 USER3
                 .STRINGZ
                            "john"
                            "xrnym"
                                             ; "smith"
 217 PW3
                 .STRINGZ
 218
 221 ; function to jump to where SUCESSS is.
 222 PWJMP LD R4, Save2 ; load back R4, which is address for YESPW.
             JMP
 223
                    R4
                                    ; jump to where YESPW is.
 224
 225 ; function for comparing input to PW1
                                ; load starting address of PW1 into R2
 226 PWCom1
            LEA
                    R2, PW1
                                    ; load address of input PW
                     R3, SPACE
 227
             LEA
                                   ; R4<-M[R2] // R4<-M[address]...store character from PW1 into R4
 228 Loop4
                    R4, R2, #0
             I DR
 229
             BRz
                     #1
                                   ; If R4 = 0 ==> NULL character, no conversion needed, skip.
 230
             ADD
                     R4, R4, #-5
                                   ; subtract by 5 as the encription increased it by 5.
                                   ; increments R2 to next address
; R5<-M[R3]...store character from input into R5
             ADD
                     R2, R2, #1
 231
 232
             LDR
                     R5, R3, #0
             ADD
                     R3, R3, #1
 233
                                   ; increments R3 to next address
 234
             NOT
                     R5, R5
 235
             ADD
                     R5, R5, #1
                                   ; 2s comp => negative
 236
             ADD
                     R6, R4, R5
                                   ; If R6 = 0 => same character
 237
             BRnp
                    NOTPW1
                                   ; not a valid username
                                    ; List R4 again for branch statement
 238
             ADD
                     R4, R4, #0
 239
             BRz
                    PWJMP
                                    ; if R4 and R5 is the NULL character, input is done, and matched with PW1
 240
             BR
                     Loop4
 241 NOTPW1
            RET
                                     ; the input was not this username, try to compare against the next one.
 242
 243
```

EECS 20 Homework Page 4 of 12

```
244 ; function for comparing input to PW2
                                   ; load starting address of PW2 into R2
245 PWCom2 LEA
                     R2, PW2
                     R3, SPACE
                                      ; load address of input PW
246
             ΙFΔ
247 Loop5
             LDR
                     R4, R2, #0
                                     ; R4<-M[R2] // R4<-M[address]...store character from PW2 into R4
248
              BRz
                      #1
                                      ; If R4 = 0 ==> NULL character, no conversion needed, skip.
                                      ; subtract by 5 as the encription increased it by 5.
249
             ADD
                     R4, R4, #-5
250
                     R2, R2, #1
                                      ; increments R2 to next address
              ADD
251
              LDR
                      R5, R3, #0
                                      ; R5<-M[R3]...store character from input into R5
252
              ADD
                     R3, R3, #1
                                      ; increments R3 to next address
             NOT
                     R5, R5
253
254
             ADD
                     R5, R5, #1
                                      ; 2s comp => negative
255
              ADD
                     R6, R4, R5
                                      ; If R6 = 0 => same character
256
              BRnp
                     NOTPW2
                                      ; not a valid username
                                      ; List R4 again for branch statement
257
              ADD
                     R4, R4, #0
                     PWJMP
258
             BRz
                                      ; if R4 and R5 is the NULL character, input is done, and matched with PW2
259
              BR
                     Loop5
260 NOTPW2
            RET
                                      ; the input was not this username, try to compare against the next one.
261
 262 ; function for comparing input to PW3
                                      ; load starting address of PW3 into R2
 263 PWCom3 LEA
                      R2, PW3
 264
              LEA
                      R3, SPACE
                                      ; load address of input PW
 265 Loop6
                                      ; R4<-M[R2] // R4<-M[address]...store character from PW3 into R4
              LDR
                      R4, R2, #0
                                      ; If R4 = 0 ==> NULL character, no conversion needed, skip.
 266
              BR<sub>7</sub>
                      #1
 267
              ADD
                      R4, R4, #-5
                                      ; subtract by 5 as the encription increased it by 5.
 268
              ADD
                      R2, R2, #1
                                      ; increments R2 to next address
                                      ; R5<-M[R3]...store character from input into R5
                      R5, R3, #0
 269
              LDR
 270
              ADD
                      R3, R3, #1
                                      ; increments R3 to next address
 271
              NOT
                      R5, R5
                      R5, R5, #1
R6, R4, R5
 272
              ADD
                                      ; 2s comp => negative
              ADD
                                      ; If R6 = 0 => same character
 273
                                      ; not a valid username
 274
              BRnp
                      NOTPW3
 275
              ADD
                      R4, R4, #0
                                      ; List R4 again for branch statement
              BRz
                      PWJMP
 276
                                      ; if R4 and R5 is the NULL character, input is done, and matched with PW3
              BR
 277
                      Loop6
 278 NOTPW3
             RET
                                      ; the input was not this username, try to compare against the next one.
 279
             . END
280
```

EECS 20 Homework Page 5 of 12

SCREENSHOT OF THE LOCATION OF THE USERNAMES & PASSWORDS (ENCRYPTED PASSWORDS)

The encrypted passwords are the ones in the code. The actual passwords are in the comments.

SCREENSHOT OF A USER LOGGING IN SUCCESSFULLY

User panteater:

EECS 20 Homework Page 6 of 12

	R	egister	's		Memory								
R0	x0000	0			0	•	x036C	x0FF9	4089	BRnzp TRAP_HALT			
R1	x7FFF	32767			0	⊩	x036D	XFFFE	65534	OS_MCR .FILL xFFFE			
R2	x3155	12629			0	⊩	x036E	x7FFF	32767	MASK_HI .FILL x7FFF			
R3	x309F	12447			0	 	x036F	X000A	10				
R4	x3030	12336			0	>	x 0370	x000A	10				
R5	x0000	0			0	▶	x0371	x002D	45	-			
R6	x2FFE	12286			0	>	x0372	x002D	45	-			
R7	x3023	12323			0	>	x 0373	x002D	45	-			
PSR	x0002	2	CC: Z		0	>	x0374	x0020	32				
PC	x036C	876			0	>	x 0375	x0048	72	Н			
MCR	x0000	0			0	>	x 0376	x0061	97	a			
				_	0	>	x 0377	x006C	108	1			
			to focus)	×	0	>	x 0378	x0074	116	t			
'No in			her priority	•	0	>	x0379	x0069	105	i			
pendin	g' sched	uled fo	or		0	▶	x037A	x006E	110	n			
-31999					0	▶	x037B	x0067	103	g			
			Skipping		0	>	x037C	x0020	32				
-	ing Keyb 19997226		scheduled		0	 	x 037D	x0074	116	t			
			Skipping		0	>	x037E	x0068	104	h			
'Updat	ing Disp	lay' s	cheduled for		0	>	x037F	x0065	101	е			
-31999					0	 	x 0380	x0020	32				
			Skipping		0	 	x0381	x004C	76	L			
	terrupt g'sched	-	her priority or		0	>	x0382	x0043	67	С			
-31999	-	wied I	-		0	>	x 0383	x002D	45	-			
Enter	your use	rname:	panteater		0	>	x0384	x0033	51	3			
	your pas		,		0	>	x0385	x0020	32				
pantea	ter. You	have :	logged in		0	>	x0386	x002D	45	-			
На	lting th	e LC-3								PC ← ← →			

User qv:

EECS 20 Homework Page 7 of 12

	R	egister	rs		Memory								
R0	x0000	0		0	•	x036C	x0FF9	4089	BRnzp TRAP_HALT				
R1	x7FFF	32767		0		x036D	XFFFE	65534	OS_MCR .FILL XFFFE				
R2	x3164	12644		0	▶	x036E	x7FFF	32767	MASK_HI .FILL x7FFF				
R3	x30A5	12453		0	▶	x036F	x000A	10					
R4	x3030	12336		0	▶	x0370	X000A	10					
R5	x0000	0		0	>	x0371	x002D	45	-				
R6	x2FFE	12286		0		x0372	x002D	45	-				
R7	x3024	12324		0		x0373	x002D	45	-				
PSR	x0002	2	CC: Z	0	 	x0374	x0020	32					
PC	x036C	876		0		x 0375	x0048	72	Н				
MCR	x0000	0		0		x 0376	x0061	97	a				
			_	0		x0377	x006C	108	1				
		•	to focus)	0		x0378	x0074	116	t				
'No in			her priority	0	 	x0379	x0069	105	i				
pendin	g' sched	uled f	or	0	 	x037A	x006E	110	n				
-29628				0	>	x037B	x0067	103	g				
			Skipping	0	 	x037C	x0020	32					
-	ing Keyb 196281646		scheduled	0		x037D	x0074	116	t				
			Skipping	0		x037E	x0068	104	h				
'Updat	ing Disp	lay's	cheduled for	0		x037F	x0065	101	е				
-29628				0		x0380	x0020	32					
			Skipping	0	 	x0381	x004C	76	L				
	iterrupt ig' sched	_	her priority	0		x0382	x0043	67	С				
-29628	_	area I	-	0		x0383	x002D	45	-				
Enter	your use	rname:	qv	0		x0384	x0033	51	3				
Enter	your pas	sword:	Hello, qv.	0		x0385	x0020	32					
You ha	ve logge	d in		0		x0386	x002D	45	-				
На	lting th	e LC-3							DO / / \				

User john

EECS 20 Homework Page 8 of 12

	R	Register	rs		Memory							
R0	x0000	0			0	•	x036C	x0FF9	4089	BRnzp TRAP_HALT		
R1	x7FFF	32767			0	⊩	x036D	XFFFE	65534	OS_MCR .FILL xFFFE		
R2	x316F	12655			0	⊩	x036E	x7FFF	32767	MASK_HI .FILL x7FFF		
R3	x309F	12447			0	⊩	x036F	x000A	10			
R4	x3030	12336			0	⊩	x 0370	x000A	10			
R5	x0000	0			0	>	x0371	x002D	45	-		
R6	x2FFE	12286			0	 	x0372	x002D	45	-		
R7	x3025	12325			0	⊩	x0373	x002D	45	-		
PSR	x0002	2	CC: Z		0	 	x0374	x0020	32			
PC	x036C	876			0	⊩	x0375	x0048	72	Н		
MCR	x0000	0			0	 	x0376	x0061	97	a		
				_	0	⊩	x0377	x006C	108	1		
			to focus)	×	0	⊩	x0378	x0074	116	t		
No in			her priority	•	0	⊩	x0379	x0069	105	í		
endin	g' sched	duled f	or		0	 	x037A	x006E	110	n		
-27006					0	 	x037B	x0067	103	g		
			Skipping		0	 	x037C	x0020	32			
-	7006810 <i>6</i>		scheduled		0	 	x037D	x0074	116	t		
			Skipping		0	 	x037E	x0068	104	h		
'Updat	ing Disp	olay's	cheduled for		0	 	x037F	x0065	101	е		
-27006	8106				0	 	x0380	x0020	32			
			Skipping		0	 	x0381	x004C	76	L		
	terrupt g'sched		her priority or		0	>	x0382	x0043	67	C		
-27006	-	WIEW I			0	 	x 0383	x002D	45	-		
Enter	your use	ername:	john		0	 	x0384	x0033	51	3		
Enter	your pas	ssword:	Hello,		0		x0385	x0020	32			
john.	You have	e logge	d in		0	 	x 0386	x002D	45	-		
На	lting th	ne LC-3								PC ← ←		

SCREENSHOT OF A USER LOGGING IN UNSUCCESSFULLY (EITHER WRONG USER NAME, WRONG PASSWORD, OR BOTH)

Testing with panteater, username is typed wrong once and password is typed wrong twice:

EECS 20 Homework Page 9 of 12

	R	egister	s		Memory								
R0	x0000	0		0		x036C	x0FF9	4089	BRnzp TRAP_HALT				
R1	x7FFF	32767		0	•	x036D	XFFFE	65534	OS_MCR .FILL xFFFE				
R2	x3155	12629		0	▶	x036E	x7FFF	32767	MASK_HI .FILL x7FFF				
R3	x309F	12447		0		x036F	x000A	10					
R4	x3030	12336		0	>	x 0370	x000A	10					
R5	x0000	0		0	•	x0371	x002D	45	-				
R6	x2FFE	12286		0		x0372	x002D	45	-				
R7	x3023	12323		0	•	x0373	x002D	45	-				
PSR	x0002	2	CC: Z	0	•	x0374	x0020	32					
PC	x036C	876		0	>	x0375	x0048	72	Н				
MCR	x0000	0		0	>	x0376	x0061	97	a				
			_	0	>	x0377	x006C	108	1				
		•	to focus)	0	>	x0378	x0074	116	t				
-	50860186			. 0	>	x0379	x0069	105	i				
	g: -2508	60179:	Skipping	0		x037A	x006E	110	n				
'Updat	ing Disp	olay' s	cheduled for	0		x037B	x0067	103	g				
-25086				0		x037C	x0020	32					
			Skipping her priority	0	>	x037D	x0074	116	t				
	g' sched	-		0		x037E	x0068	104	h				
-25086	-			0		x037F	x0065	101	e				
Enter	your use	rname:	pant	0		x0380	x0020	32					
		-	Enter your	0		x0381	x004C	76	L				
	me: pant		Na+1:-1	0	>	x0382	x0043	67	C				
			Not valid, password:	0	•	x0383	x002D	45	-				
		-	Enter your	0		x0384		51	3				
		-	teater. You	0		×0385							
have l	ogged in	L		0		x0386			-				
На	lting th	e LC-3							PC ← ←				

Testing with qv, username typed in wrong twice, password is right first time:

EECS 20 Homework Page 10 of 12

	R	egister	s						Mer	nory
R0	x0000	0			0		x036C	x0FF9	4089	BRnzp TRAP_HALT
R1	x7FFF	32767			0	▶	x036D	XFFFE	65534	OS_MCR .FILL xFFFE
R2	x3164	12644			0	▶	x036E	x7FFF	32767	MASK_HI .FILL x7FFF
R3	x30A5	12453			0	 	x036F	X000A	10	
R4	x3030	12336			0	 	x 0370	X000A	10	
R5	x0000	0			0	 	x0371	x002D	45	-
R6	x2FFE	12286			0	 	x0372	x002D	45	-
R7	x3024	12324			0	 	x0373	x002D	45	-
PSR	x0002	2	CC: Z		0	 	x0374	x0020	32	
PC	x036C	876			0		x 0375	x0048	72	Н
MCR	x0000	0			0	>	x0376	x0061	97	а
					0		x0377	x006C	108	1
Console (click to focus)							0270	x0074	116	t.
		•		_	0	-	XU3/0	XUU/4	TT0	L
	y. 1007		nuthbrind	_	0		x0376			i
Updat	y. 1007	oard'		^				x0069	105	
Updat	ing Keyb 35711066	oard':	nuthbrind	^	0	>	x0379	x0069 x006E	105 110	i
Updat For -1 varnin Updat	ing Keyb 35711066 g: -1357	oard' : : :11059:	scheduled	•	0	>	x0379 x037A	x0069 x006E x0067	105 110 103	i n
Updat For -1 Varnin Updat	ing Keyb 35711066 g: -1357 ing Disp 1066	ooard'; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	scheduled Skipping	•	0	> > > >	x0379 x037A x037B	x0069 x006E x0067 x0020	105 110 103 32	i n
Updat For -1 Varnin Updat -13571	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357	oard': 11059: lay's:	scheduled Skipping cheduled i	for	9	>>>	x0379 x037A x037B x037C	x0069 x006E x0067 x0020 x0074	105 110 103 32 116	i n g
Updat For -1 Varnin Updat -13571 Varnin	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357	oard'; 11059: lay's 11059: of high	Skipping cheduled in Skipping ker priori	for	0	>>>>	x0379 x037A x037B x037C x037D	x0069 x006E x0067 x0020 x0074 x0068	105 110 103 32 116 104	i n g
Updat For -1 Varnin Updat -13571 Varnin	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357 terrupt g' sched	oard'; 11059: lay's 11059: of high	Skipping cheduled in Skipping ker priori	for	9	> > > > >	x0379 x037A x037B x037C x037D	x0069 x006E x0067 x0020 x0074 x0068 x0065	105 110 103 32 116 104	i n g t
Updat For -1 Varnin Updat -13571 Varnin No in bendin -13571	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357 terrupt g' sched	oard'; 11059: lay's: 11059: of high	Skipping cheduled Skipping cheduled Skipping her priori	for	0 0 0 0 0 0	>>>>	x0379 x037A x037B x037C x037D x037E x037F x0380	x0069 x006E x0067 x0020 x0074 x0068 x0065	105 110 103 32 116 104 101	i n g t h
Updat For -1 Varnin Updat -13571 Varnin No in Dendin -13571 Enter	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357 terrupt g' sched 1065 your use lid, try	onard' : 11059: clay' so 11059: of high	Skipping cheduled Skipping cheduled Skipping her priori	for	0 0 0 0 0 0 0 0 0	>>>>>	x0379 x037A x037B x037C x037D x037E x037F x0380 x0381	x0069 x006E x0067 x0020 x0074 x0068 x0065 x0020 x004C	105 110 103 32 116 104 101 32	i n g t h
Updat For -1 Varnin Updat -13571 Varnin No in bendin -13571 Enter Vot va	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357 terrupt g' sched 1065 your use lid, try me: qvc	lay's of high	Skipping cheduled is Skipping her priori or qa Enter you	for	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	> > > > > > >	x0379 x037A x037B x037C x037D x037E x037F x0380 x0381 x0382	x0069 x006E x0067 x0020 x0074 x0068 x0065 x0020 x004C	105 110 103 32 116 104 101 32 76	i n g t h
Updat For -1 Varnin Updat -13571 Varnin No in Dendin -13571 Enter Not va	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357 terrupt g' sched 1065 your use lid, try me: qvc lid, try	lay's of high	Skipping cheduled is Skipping her priorior	for	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	>>>>>>	x0379 x037A x037B x037C x037D x037E x037F x0380 x0381 x0382 x0383	x0069 x006E x0067 x0020 x0074 x0068 x0065 x0020 x004C x0043	105 110 103 32 116 104 101 32 76 67 45	i n g t h e C -
Updat For -1 Varnin Updat -13571 Varnin No in Dendin -13571 Enter Not va userna Vot va	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357 terrupt g' sched 1065 your use lid, try me: qvc lid, try me: qv	onard': 11059: lay'se 11059: of high uled for	Skipping cheduled is Skipping her priori or qa Enter you	ior .ty	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	> > > > <	x0379 x037A x037B x037C x037D x037E x037F x0380 x0381 x0382 x0383	x0069 x006E x0067 x0020 x0074 x0068 x0065 x0020 x004C x0043 x002D	105 110 103 32 116 104 101 32 76 67 45	i n g t h e L C
Updat For -1 Varnin Updat -13571 Varnin No in Dendin -13571 Enter Not va userna Vot va userna Enter	ing Keyb 35711066 g: -1357 ing Disp 1066 g: -1357 terrupt g' sched 1065 your use lid, try me: qvc lid, try me: qv	coard': 11059: lay's 11059: of high uled for rname: again again sword:	scheduled Skipping cheduled i Skipping her priori or qa Enter you	ior .ty	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	> > > > <	x0379 x037A x037B x037C x037D x037E x037F x0380 x0381 x0382 x0383	x0069 x006E x0067 x0020 x0074 x0068 x0065 x0020 x004C x0043 x002D x0033	105 110 103 32 116 104 101 32 76 67 45 51	i n g t h e C -

Testing with john, username right, password wrong once:

EECS 20 Homework Page 11 of 12

	R	egister	rs .		Memory									
R0	x0000	0			0	•	x036C	x0FF9	4089	BRnzp TRAP_HALT				
R1	x7FFF	32767			0	⊩	x036D	XFFFE	65534	OS_MCR .FILL xFFFE				
R2	x316F	12655			0	⊩	x036E	x7FFF	32767	MASK_HI .FILL x7FFF				
R3	x309F	12447			0	⊩	x036F	x000A	10					
R4	x3030	12336			0	⊩	x 0370	x000A	10					
R5	x0000	0			0	▶	x0371	x002D	45	-				
R6	x2FFE	12286			0	 	x0372	x002D	45	-				
R7	x3025	12325			0	 	x 0373	x002D	45	-				
PSR	x0002	2	CC: Z		0	 	x0374	x0020	32					
PC	x036C	876			0	 	x0375	x0048	72	Н				
MCR	x0000	0			0	 	x0376	x0061	97	а				
						 	x0377	x006C	108	1				
J1007		(click t	to focus)	×	0	 	x0378	x0074	116	t				
		4124:	Skipping 'No	•	0	 	x0379	x0069	105	i				
			priority		0	 	x037A	x006E	110	n				
pendin	g' sched	luled f	or -51864125		0	 	x037B	x0067	103	g				
			Skipping		0	 	x037C	x0020	32					
-	ing Keyb 1863206	oard'	scheduled		0	 	x037D	x0074	116	t				
		3199:	Skipping		0	 	x037E	x0068	104	h				
			cheduled for		0	 	x037F	x0065	101	e				
-51863	206				0		x0380	x0020	32					
			Skipping 'No		0	 	x0381	x004C	76	L				
	-		priority		0	<u> </u>	x0382	x0043	67	C				
•	your use		or -51863205		0	<u> </u>	x0383	x002D	4.5	-				
	_		Not valid,		0		x0384			3				
			password:		0		x0385							
Hello,	john. Y	ou hav	e logged in		0		x0386			_				
На	lting th	e LC-3					22200			PC ← ← →				

EECS 20 Homework Page 12 of 12