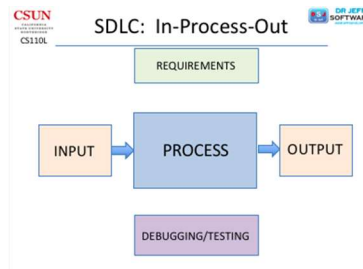


Lab #: 4 Software Interrupts

LAB FORM



Requirements

1. Decode Pending Interrupts
2. Allocate memory for Handlers
3. Use Jump Table
 - a) Order by Priority
 - b) Test and Jump
 - c) Handlers as subroutines: `jal -> jr $ra`

Inputs

paste here screenshots of all inputs (not code): .data segment & GUI/console

Data Segment: (Hexadecimal Addresses)

| Data Segment | | | | | | | | |
|--------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
| 0x10010000 | 1986356003 | 543318348 | 1226848820 | 1919251566 | 1953527154 | 2036473971 | 1936673312 | 543257960 |
| 0x10010020 | 2627 | 1702129221 | 1850286194 | 1496855556 | 540689744 | 1296973104 | 824192073 | 1230392893 |
| 0x10010040 | 1026695212 | 539773270 | 1632124217 | 29804 | 1702129221 | 1850286194 | 1700143220 | 1919906915 |
| 0x10010060 | 758130720 | 975779121 | 0 | 541674830 | 1702129257 | 1896745202 | 1308631412 | 1763723606 |
| 0x10010080 | 1919251566 | 1953527154 | 1444937761 | 1852383305 | 1920099700 | 561279093 | 1667585536 | 1030909812 |
| 0x100100a0 | 0 | 1869771333 | 1763719794 | 1734700140 | 1226861665 | 1411413102 | 6679429 | 1869771333 |
| 0x100100c0 | 1763719794 | 1734700140 | 1696623713 | 2037544046 | 0 | 1953259848 | 539059301 | 1685024853 |
| 0x100100e0 | 1702453805 | 0 | 1886352467 | 543450480 | 561280367 | 0 | 707406372 | 707406372 |
| 0x10010100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x10010120 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x10010140 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x10010160 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x10010180 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x100101a0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII

(ASCII):

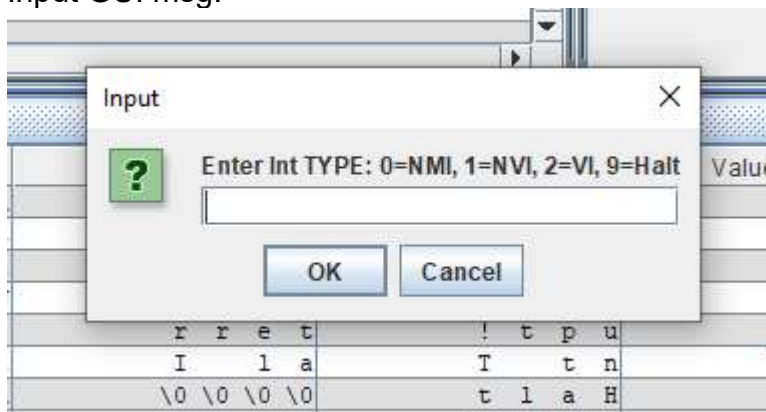
| Data Segment | | | | | | | | | |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--|
| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) | |
| 0x10010000 | v e c # | b a L | I : 4 | r e t n | t p u r | y b s | s o J | a u h | |
| 0x10010020 | \0 \0 \n C | e t n E | n I : r | Y T t | : E P | M N = 0 | l , I | I V N = | |
| 0x10010040 | = 2 , | , I V | a H = 9 | \0 \0 t l | e t n E | n I r | e V t | r o t c | |
| 0x10010060 | - 0 (| :) 5 l | \0 \0 \0 \0 | I M N | e t n i | p u r r | N \0 : t | i I V | |
| 0x10010080 | r e t n | t p u r | V \0 ! | n i I | r r e t | t t p u | c e V \0 | = r o t | |
| 0x100100a0 | \0 \0 \0 \0 | o r r E | i : r | g e l l | I l a | T t n | \0 e p y | o r r E | |
| 0x100100c0 | i : r | g e l l | e l a | y r t n | \0 \0 \0 \0 | t l a H | ! d e | d o o g | |
| 0x100100e0 | e y b - | \0 \0 \0 \0 | p o t S | d e p | ! t u o | \0 \0 \0 \0 | * * * g | * * * g | |
| 0x10010100 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | |
| 0x10010120 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | |
| 0x10010140 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | |
| 0x10010160 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | |
| 0x10010180 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | |
| 0x100101a0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | |

0x10010000 (.data)
☒ Hexadecimal Addresses
 ☐ Hexadecimal Values
 ☒ ASCII

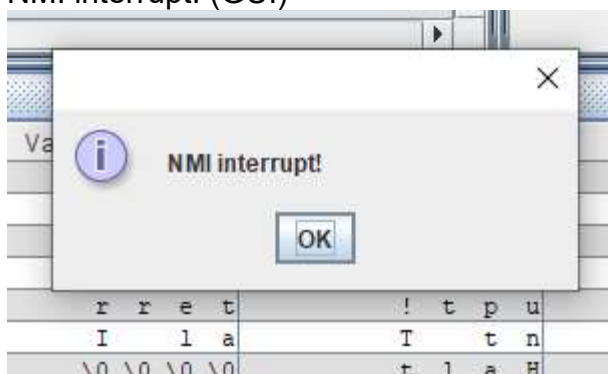
Outputs (note: you must execute the code to get outputs)

paste here screenshots of all outputs (not code): console/GUI, .data, .heap, registers

Input GUI msg:



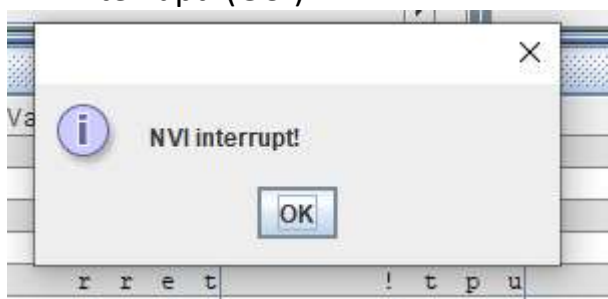
NMI interrupt: (GUI)



(Console):



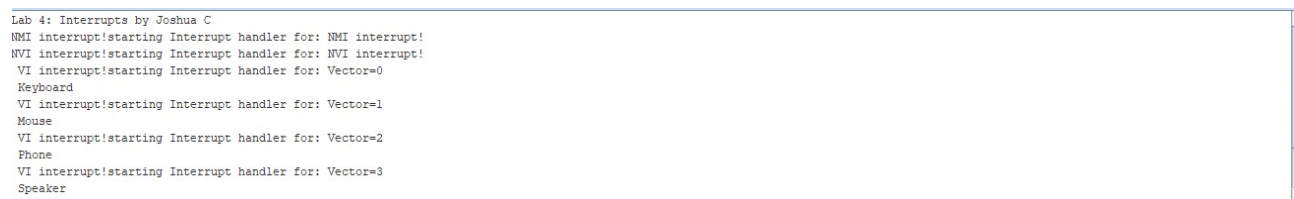
NVI interrupt: (GUI)



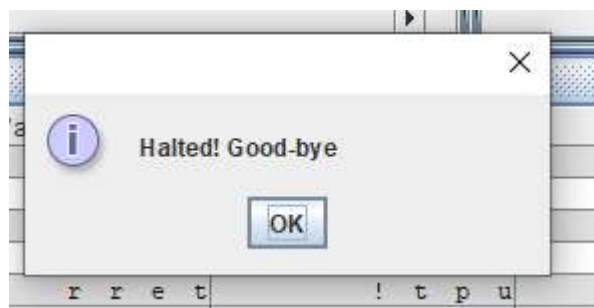
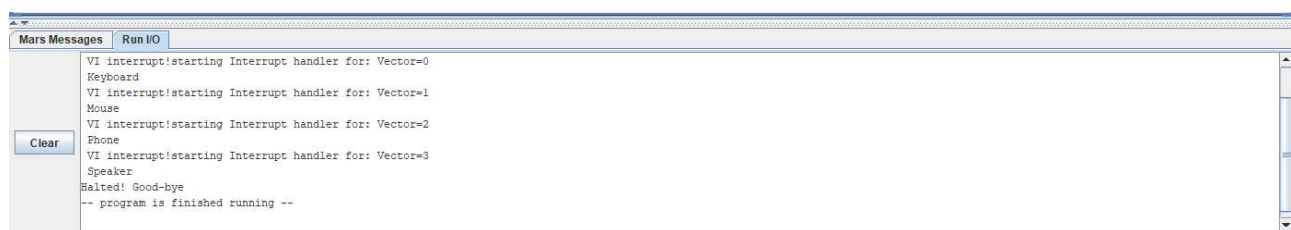
(Console):



VI Vector Interrupt: Console Vectors Outputs (V0-V3):



Halt: Console & GUI Halt msg:



Data Segment (Output) ASCII:

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 0x10010000 | v e c # | b a L | I : 4 | t e t n | t p u r | y b a | a o J | a u h |
| 0x10010020 | \0 \0 \n C | e t n E | n I r | Y T t | : E P | M N = 0 | i , I | I V N = |
| 0x10010040 | = 2 , | , I V | a H = 9 | \0 \0 t l | e t n E | n I r | e v t | r o t c |
| 0x10010060 | - 0 (| :) 5 l | \0 \0 \0 \0 | I M N | e t n i | p u r r | N \0 ! t | i I V |
| 0x10010080 | r e t n | t p u r | V \0 ! | n i I | r r e t | ! t p u | c e V \0 | = r o t |
| 0x100100a0 | \0 \0 \0 \0 | o r r E | i : r | g e l l | I l a | T t n | \0 e p y | o r r E |
| 0x100100c0 | i : r | g e l l | e l a | y r t n | \0 \0 \0 \0 | t l a H | ! d e | d o o G |
| 0x100100e0 | e y b - | \0 \0 \0 \0 | p o t S | d e p | ! t u o | \0 \0 \0 \0 | * * * s | * * * s |
| 0x10010100 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010120 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010140 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010160 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x10010180 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |
| 0x100101a0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 | \0 \0 \0 \0 |

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☒ ASCII

Skip to next page for source
code



Process (algorithms): *source code**(paste original syntax colored view here; include line numbers – make legible)*

```

1  ##Lab 4 -- Interrupts
2  #by Joshua Cohen
3  #Version: 1.1 > 4.27.21
4  #Interrupt vectors:
5  #0-3 active
6  #reg map: $t9 = stop val; $s0, $s1 for int vectors
7  .data
8  vector:.asciiz "\x00" #reserve 4 bytes
9  header:.asciiz "Lab 4: Interrupts By Joshua C\n"
10 .align 2
11 prompt1:.asciiz "Enter Int TYPE: 0=NMI, 1=NVI, 2=VI, 9=Halt"
12 .align 2
13 prompt2:.asciiz "Enter Int Vector (0-15):"
14 .align 2
15 NMI_str:.asciiz "NMI interrupt!"
16 NVI_str:.asciiz "NVI interrupt!"
17 VI_str:.asciiz "VI interrupt!"
18 Vect_str:.asciiz "Vector="
19 .align 2
20 Err_msg1:.asciiz "Error: illegal Int Type"
21 .align 2
22 Err_msg2:.asciiz "Error: illegal entry"
23 .align 2
24 Halt_msg:.asciiz "Halted! Good-bye"
25 .align 2
26 Stop_msg:.asciiz "Stopped out!"
27 .align 2
28 end_data:.asciiz "*****"
29 #define
30 .eqv heap, 0x10040000
31 .eqv in_buf, 0x10040020 #input buffer
32 .eqv exc_seg, 0x80000180
33 .eqv stop, 10
34 #macros
35 .macro done
36     li $v0, 10 #stop code
37     syscall #stop
38 .end_macro
39 .macro print_mac (%str)
40     la $a0, %str
41     li $v0, 4
42     syscall
43 .end_macro
44 .macro msgbox (%str)
45     la $a0, %str
46     li $v0, 55 #GUI msg code
47     li $a1, 1 #msg type is info
48     syscall
49 .end_macro
50
51 ***ISR macro-> Trap
52 .macro _ISR (%str)
53     la $a0, %str
54     jal printStr
55     jal GUI_out
56     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
57     b loop_main
58 .end_macro
59 #code
60 .text
61 #saved:
62 $t9
63 li $t9, stop
64 la $a0, header
65 jal printStr
66 #main Loop
67 loop_main:
68     subiu $t9, $t9, 1 #decrement counter
69     blez $t9, Stop
70     la $a0, prompt1
71     jal printStr
72     li $v0, 1
73     syscall
74     la $a0, prompt2
75     jal printStr
76     li $v0, 1
77     syscall
78     la $a0, Vect_str
79     jal printStr
80     li $v0, 1
81     syscall
82     la $a0, Err_msg1
83     jal printStr
84     li $v0, 1
85     syscall
86     la $a0, Err_msg2
87     jal printStr
88     li $v0, 1
89     syscall
90     la $a0, Halt_msg
91     jal printStr
92     li $v0, 1
93     syscall
94     la $a0, Stop_msg
95     jal printStr
96     li $v0, 1
97     syscall
98     la $a0, end_data
99     jal printStr
100    li $v0, 1
101    syscall
102    .macro done
103        li $v0, 10 #stop code
104        syscall #stop
105    .end_macro
106    .macro print_mac (%str)
107        la $a0, %str
108        li $v0, 4
109        syscall
110    .end_macro
111    .macro msgbox (%str)
112        la $a0, %str
113        li $v0, 55 #GUI msg code
114        li $a1, 1 #msg type is info
115        syscall
116    .end_macro
117
118 ***ISR macro-> Trap
119 .macro _ISR (%str)
120     la $a0, %str
121     jal printStr
122     jal GUI_out
123     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
124     b loop_main
125 .end_macro
126 #code
127 .text
128 #saved:
129 $t9
130 li $t9, stop
131 la $a0, header
132 jal printStr
133 #main Loop
134 loop_main:
135     subiu $t9, $t9, 1 #decrement counter
136     blez $t9, Stop
137     la $a0, prompt1
138     jal printStr
139     li $v0, 1
140     syscall
141     la $a0, prompt2
142     jal printStr
143     li $v0, 1
144     syscall
145     la $a0, Vect_str
146     jal printStr
147     li $v0, 1
148     syscall
149     la $a0, Err_msg1
150     jal printStr
151     li $v0, 1
152     syscall
153     la $a0, Err_msg2
154     jal printStr
155     li $v0, 1
156     syscall
157     la $a0, Halt_msg
158     jal printStr
159     li $v0, 1
160     syscall
161     la $a0, Stop_msg
162     jal printStr
163     li $v0, 1
164     syscall
165     la $a0, end_data
166     jal printStr
167     li $v0, 1
168     syscall
169     .macro done
170         li $v0, 10 #stop code
171         syscall #stop
172     .end_macro
173     .macro print_mac (%str)
174         la $a0, %str
175         li $v0, 4
176         syscall
177     .end_macro
178     .macro msgbox (%str)
179         la $a0, %str
180         li $v0, 55 #GUI msg code
181         li $a1, 1 #msg type is info
182         syscall
183     .end_macro
184
185 ***ISR macro-> Trap
186 .macro _ISR (%str)
187     la $a0, %str
188     jal printStr
189     jal GUI_out
190     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
191     b loop_main
192 .end_macro
193 #code
194 .text
195 #saved:
196 $t9
197 li $t9, stop
198 la $a0, header
199 jal printStr
200 #main Loop
201 loop_main:
202     subiu $t9, $t9, 1 #decrement counter
203     blez $t9, Stop
204     la $a0, prompt1
205     jal printStr
206     li $v0, 1
207     syscall
208     la $a0, prompt2
209     jal printStr
210     li $v0, 1
211     syscall
212     la $a0, Vect_str
213     jal printStr
214     li $v0, 1
215     syscall
216     la $a0, Err_msg1
217     jal printStr
218     li $v0, 1
219     syscall
220     la $a0, Err_msg2
221     jal printStr
222     li $v0, 1
223     syscall
224     la $a0, Halt_msg
225     jal printStr
226     li $v0, 1
227     syscall
228     la $a0, Stop_msg
229     jal printStr
230     li $v0, 1
231     syscall
232     la $a0, end_data
233     jal printStr
234     li $v0, 1
235     syscall
236     .macro done
237         li $v0, 10 #stop code
238         syscall #stop
239     .end_macro
240     .macro print_mac (%str)
241         la $a0, %str
242         li $v0, 4
243         syscall
244     .end_macro
245     .macro msgbox (%str)
246         la $a0, %str
247         li $v0, 55 #GUI msg code
248         li $a1, 1 #msg type is info
249         syscall
250     .end_macro
251
252 ***ISR macro-> Trap
253 .macro _ISR (%str)
254     la $a0, %str
255     jal printStr
256     jal GUI_out
257     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
258     b loop_main
259 .end_macro
260 #code
261 .text
262 #saved:
263 $t9
264 li $t9, stop
265 la $a0, header
266 jal printStr
267 #main Loop
268 loop_main:
269     subiu $t9, $t9, 1 #decrement counter
270     blez $t9, Stop
271     la $a0, prompt1
272     jal printStr
273     li $v0, 1
274     syscall
275     la $a0, prompt2
276     jal printStr
277     li $v0, 1
278     syscall
279     la $a0, Vect_str
280     jal printStr
281     li $v0, 1
282     syscall
283     la $a0, Err_msg1
284     jal printStr
285     li $v0, 1
286     syscall
287     la $a0, Err_msg2
288     jal printStr
289     li $v0, 1
290     syscall
291     la $a0, Halt_msg
292     jal printStr
293     li $v0, 1
294     syscall
295     la $a0, Stop_msg
296     jal printStr
297     li $v0, 1
298     syscall
299     la $a0, end_data
300     jal printStr
301     li $v0, 1
302     syscall
303     .macro done
304         li $v0, 10 #stop code
305         syscall #stop
306     .end_macro
307     .macro print_mac (%str)
308         la $a0, %str
309         li $v0, 4
310         syscall
311     .end_macro
312     .macro msgbox (%str)
313         la $a0, %str
314         li $v0, 55 #GUI msg code
315         li $a1, 1 #msg type is info
316         syscall
317     .end_macro
318
319 ***ISR macro-> Trap
320 .macro _ISR (%str)
321     la $a0, %str
322     jal printStr
323     jal GUI_out
324     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
325     b loop_main
326 .end_macro
327 #code
328 .text
329 #saved:
330 $t9
331 li $t9, stop
332 la $a0, header
333 jal printStr
334 #main Loop
335 loop_main:
336     subiu $t9, $t9, 1 #decrement counter
337     blez $t9, Stop
338     la $a0, prompt1
339     jal printStr
340     li $v0, 1
341     syscall
342     la $a0, prompt2
343     jal printStr
344     li $v0, 1
345     syscall
346     la $a0, Vect_str
347     jal printStr
348     li $v0, 1
349     syscall
350     la $a0, Err_msg1
351     jal printStr
352     li $v0, 1
353     syscall
354     la $a0, Err_msg2
355     jal printStr
356     li $v0, 1
357     syscall
358     la $a0, Halt_msg
359     jal printStr
360     li $v0, 1
361     syscall
362     la $a0, Stop_msg
363     jal printStr
364     li $v0, 1
365     syscall
366     la $a0, end_data
367     jal printStr
368     li $v0, 1
369     syscall
370     .macro done
371         li $v0, 10 #stop code
372         syscall #stop
373     .end_macro
374     .macro print_mac (%str)
375         la $a0, %str
376         li $v0, 4
377         syscall
378     .end_macro
379     .macro msgbox (%str)
380         la $a0, %str
381         li $v0, 55 #GUI msg code
382         li $a1, 1 #msg type is info
383         syscall
384     .end_macro
385
386 ***ISR macro-> Trap
387 .macro _ISR (%str)
388     la $a0, %str
389     jal printStr
390     jal GUI_out
391     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
392     b loop_main
393 .end_macro
394 #code
395 .text
396 #saved:
397 $t9
398 li $t9, stop
399 la $a0, header
400 jal printStr
401 #main Loop
402 loop_main:
403     subiu $t9, $t9, 1 #decrement counter
404     blez $t9, Stop
405     la $a0, prompt1
406     jal printStr
407     li $v0, 1
408     syscall
409     la $a0, prompt2
410     jal printStr
411     li $v0, 1
412     syscall
413     la $a0, Vect_str
414     jal printStr
415     li $v0, 1
416     syscall
417     la $a0, Err_msg1
418     jal printStr
419     li $v0, 1
420     syscall
421     la $a0, Err_msg2
422     jal printStr
423     li $v0, 1
424     syscall
425     la $a0, Halt_msg
426     jal printStr
427     li $v0, 1
428     syscall
429     la $a0, Stop_msg
430     jal printStr
431     li $v0, 1
432     syscall
433     la $a0, end_data
434     jal printStr
435     li $v0, 1
436     syscall
437     .macro done
438         li $v0, 10 #stop code
439         syscall #stop
440     .end_macro
441     .macro print_mac (%str)
442         la $a0, %str
443         li $v0, 4
444         syscall
445     .end_macro
446     .macro msgbox (%str)
447         la $a0, %str
448         li $v0, 55 #GUI msg code
449         li $a1, 1 #msg type is info
450         syscall
451     .end_macro
452
453 ***ISR macro-> Trap
454 .macro _ISR (%str)
455     la $a0, %str
456     jal printStr
457     jal GUI_out
458     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
459     b loop_main
460 .end_macro
461 #code
462 .text
463 #saved:
464 $t9
465 li $t9, stop
466 la $a0, header
467 jal printStr
468 #main Loop
469 loop_main:
470     subiu $t9, $t9, 1 #decrement counter
471     blez $t9, Stop
472     la $a0, prompt1
473     jal printStr
474     li $v0, 1
475     syscall
476     la $a0, prompt2
477     jal printStr
478     li $v0, 1
479     syscall
480     la $a0, Vect_str
481     jal printStr
482     li $v0, 1
483     syscall
484     la $a0, Err_msg1
485     jal printStr
486     li $v0, 1
487     syscall
488     la $a0, Err_msg2
489     jal printStr
490     li $v0, 1
491     syscall
492     la $a0, Halt_msg
493     jal printStr
494     li $v0, 1
495     syscall
496     la $a0, Stop_msg
497     jal printStr
498     li $v0, 1
499     syscall
500     la $a0, end_data
501     jal printStr
502     li $v0, 1
503     syscall
504     .macro done
505         li $v0, 10 #stop code
506         syscall #stop
507     .end_macro
508     .macro print_mac (%str)
509         la $a0, %str
510         li $v0, 4
511         syscall
512     .end_macro
513     .macro msgbox (%str)
514         la $a0, %str
515         li $v0, 55 #GUI msg code
516         li $a1, 1 #msg type is info
517         syscall
518     .end_macro
519
520 ***ISR macro-> Trap
521 .macro _ISR (%str)
522     la $a0, %str
523     jal printStr
524     jal GUI_out
525     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
526     b loop_main
527 .end_macro
528 #code
529 .text
530 #saved:
531 $t9
532 li $t9, stop
533 la $a0, header
534 jal printStr
535 #main Loop
536 loop_main:
537     subiu $t9, $t9, 1 #decrement counter
538     blez $t9, Stop
539     la $a0, prompt1
540     jal printStr
541     li $v0, 1
542     syscall
543     la $a0, prompt2
544     jal printStr
545     li $v0, 1
546     syscall
547     la $a0, Vect_str
548     jal printStr
549     li $v0, 1
550     syscall
551     la $a0, Err_msg1
552     jal printStr
553     li $v0, 1
554     syscall
555     la $a0, Err_msg2
556     jal printStr
557     li $v0, 1
558     syscall
559     la $a0, Halt_msg
560     jal printStr
561     li $v0, 1
562     syscall
563     la $a0, Stop_msg
564     jal printStr
565     li $v0, 1
566     syscall
567     la $a0, end_data
568     jal printStr
569     li $v0, 1
570     syscall
571     .macro done
572         li $v0, 10 #stop code
573         syscall #stop
574     .end_macro
575     .macro print_mac (%str)
576         la $a0, %str
577         li $v0, 4
578         syscall
579     .end_macro
580     .macro msgbox (%str)
581         la $a0, %str
582         li $v0, 55 #GUI msg code
583         li $a1, 1 #msg type is info
584         syscall
585     .end_macro
586
587 ***ISR macro-> Trap
588 .macro _ISR (%str)
589     la $a0, %str
590     jal printStr
591     jal GUI_out
592     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
593     b loop_main
594 .end_macro
595 #code
596 .text
597 #saved:
598 $t9
599 li $t9, stop
600 la $a0, header
601 jal printStr
602 #main Loop
603 loop_main:
604     subiu $t9, $t9, 1 #decrement counter
605     blez $t9, Stop
606     la $a0, prompt1
607     jal printStr
608     li $v0, 1
609     syscall
610     la $a0, prompt2
611     jal printStr
612     li $v0, 1
613     syscall
614     la $a0, Vect_str
615     jal printStr
616     li $v0, 1
617     syscall
618     la $a0, Err_msg1
619     jal printStr
620     li $v0, 1
621     syscall
622     la $a0, Err_msg2
623     jal printStr
624     li $v0, 1
625     syscall
626     la $a0, Halt_msg
627     jal printStr
628     li $v0, 1
629     syscall
630     la $a0, Stop_msg
631     jal printStr
632     li $v0, 1
633     syscall
634     la $a0, end_data
635     jal printStr
636     li $v0, 1
637     syscall
638     .macro done
639         li $v0, 10 #stop code
640         syscall #stop
641     .end_macro
642     .macro print_mac (%str)
643         la $a0, %str
644         li $v0, 4
645         syscall
646     .end_macro
647     .macro msgbox (%str)
648         la $a0, %str
649         li $v0, 55 #GUI msg code
650         li $a1, 1 #msg type is info
651         syscall
652     .end_macro
653
654 ***ISR macro-> Trap
655 .macro _ISR (%str)
656     la $a0, %str
657     jal printStr
658     jal GUI_out
659     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
660     b loop_main
661 .end_macro
662 #code
663 .text
664 #saved:
665 $t9
666 li $t9, stop
667 la $a0, header
668 jal printStr
669 #main Loop
670 loop_main:
671     subiu $t9, $t9, 1 #decrement counter
672     blez $t9, Stop
673     la $a0, prompt1
674     jal printStr
675     li $v0, 1
676     syscall
677     la $a0, prompt2
678     jal printStr
679     li $v0, 1
680     syscall
681     la $a0, Vect_str
682     jal printStr
683     li $v0, 1
684     syscall
685     la $a0, Err_msg1
686     jal printStr
687     li $v0, 1
688     syscall
689     la $a0, Err_msg2
690     jal printStr
691     li $v0, 1
692     syscall
693     la $a0, Halt_msg
694     jal printStr
695     li $v0, 1
696     syscall
697     la $a0, Stop_msg
698     jal printStr
699     li $v0, 1
700     syscall
701     la $a0, end_data
702     jal printStr
703     li $v0, 1
704     syscall
705     .macro done
706         li $v0, 10 #stop code
707         syscall #stop
708     .end_macro
709     .macro print_mac (%str)
710         la $a0, %str
711         li $v0, 4
712         syscall
713     .end_macro
714     .macro msgbox (%str)
715         la $a0, %str
716         li $v0, 55 #GUI msg code
717         li $a1, 1 #msg type is info
718         syscall
719     .end_macro
720
721 ***ISR macro-> Trap
722 .macro _ISR (%str)
723     la $a0, %str
724     jal printStr
725     jal GUI_out
726     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
727     b loop_main
728 .end_macro
729 #code
730 .text
731 #saved:
732 $t9
733 li $t9, stop
734 la $a0, header
735 jal printStr
736 #main Loop
737 loop_main:
738     subiu $t9, $t9, 1 #decrement counter
739     blez $t9, Stop
740     la $a0, prompt1
741     jal printStr
742     li $v0, 1
743     syscall
744     la $a0, prompt2
745     jal printStr
746     li $v0, 1
747     syscall
748     la $a0, Vect_str
749     jal printStr
750     li $v0, 1
751     syscall
752     la $a0, Err_msg1
753     jal printStr
754     li $v0, 1
755     syscall
756     la $a0, Err_msg2
757     jal printStr
758     li $v0, 1
759     syscall
760     la $a0, Halt_msg
761     jal printStr
762     li $v0, 1
763     syscall
764     la $a0, Stop_msg
765     jal printStr
766     li $v0, 1
767     syscall
768     la $a0, end_data
769     jal printStr
770     li $v0, 1
771     syscall
772     .macro done
773         li $v0, 10 #stop code
774         syscall #stop
775     .end_macro
776     .macro print_mac (%str)
777         la $a0, %str
778         li $v0, 4
779         syscall
780     .end_macro
781     .macro msgbox (%str)
782         la $a0, %str
783         li $v0, 55 #GUI msg code
784         li $a1, 1 #msg type is info
785         syscall
786     .end_macro
787
788 ***ISR macro-> Trap
789 .macro _ISR (%str)
790     la $a0, %str
791     jal printStr
792     jal GUI_out
793     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
794     b loop_main
795 .end_macro
796 #code
797 .text
798 #saved:
799 $t9
800 li $t9, stop
801 la $a0, header
802 jal printStr
803 #main Loop
804 loop_main:
805     subiu $t9, $t9, 1 #decrement counter
806     blez $t9, Stop
807     la $a0, prompt1
808     jal printStr
809     li $v0, 1
810     syscall
811     la $a0, prompt2
812     jal printStr
813     li $v0, 1
814     syscall
815     la $a0, Vect_str
816     jal printStr
817     li $v0, 1
818     syscall
819     la $a0, Err_msg1
820     jal printStr
821     li $v0, 1
822     syscall
823     la $a0, Err_msg2
824     jal printStr
825     li $v0, 1
826     syscall
827     la $a0, Halt_msg
828     jal printStr
829     li $v0, 1
830     syscall
831     la $a0, Stop_msg
832     jal printStr
833     li $v0, 1
834     syscall
835     la $a0, end_data
836     jal printStr
837     li $v0, 1
838     syscall
839     .macro done
840         li $v0, 10 #stop code
841         syscall #stop
842     .end_macro
843     .macro print_mac (%str)
844         la $a0, %str
845         li $v0, 4
846         syscall
847     .end_macro
848     .macro msgbox (%str)
849         la $a0, %str
850         li $v0, 55 #GUI msg code
851         li $a1, 1 #msg type is info
852         syscall
853     .end_macro
854
855 ***ISR macro-> Trap
856 .macro _ISR (%str)
857     la $a0, %str
858     jal printStr
859     jal GUI_out
860     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
861     b loop_main
862 .end_macro
863 #code
864 .text
865 #saved:
866 $t9
867 li $t9, stop
868 la $a0, header
869 jal printStr
870 #main Loop
871 loop_main:
872     subiu $t9, $t9, 1 #decrement counter
873     blez $t9, Stop
874     la $a0, prompt1
875     jal printStr
876     li $v0, 1
877     syscall
878     la $a0, prompt2
879     jal printStr
880     li $v0, 1
881     syscall
882     la $a0, Vect_str
883     jal printStr
884     li $v0, 1
885     syscall
886     la $a0, Err_msg1
887     jal printStr
888     li $v0, 1
889     syscall
890     la $a0, Err_msg2
891     jal printStr
892     li $v0, 1
893     syscall
894     la $a0, Halt_msg
895     jal printStr
896     li $v0, 1
897     syscall
898     la $a0, Stop_msg
899     jal printStr
900     li $v0, 1
901     syscall
902     la $a0, end_data
903     jal printStr
904     li $v0, 1
905     syscall
906     .macro done
907         li $v0, 10 #stop code
908         syscall #stop
909     .end_macro
910     .macro print_mac (%str)
911         la $a0, %str
912         li $v0, 4
913         syscall
914     .end_macro
915     .macro msgbox (%str)
916         la $a0, %str
917         li $v0, 55 #GUI msg code
918         li $a1, 1 #msg type is info
919         syscall
920     .end_macro
921
922 ***ISR macro-> Trap
923 .macro _ISR (%str)
924     la $a0, %str
925     jal printStr
926     jal GUI_out
927     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
928     b loop_main
929 .end_macro
930 #code
931 .text
932 #saved:
933 $t9
934 li $t9, stop
935 la $a0, header
936 jal printStr
937 #main Loop
938 loop_main:
939     subiu $t9, $t9, 1 #decrement counter
940     blez $t9, Stop
941     la $a0, prompt1
942     jal printStr
943     li $v0, 1
944     syscall
945     la $a0, prompt2
946     jal printStr
947     li $v0, 1
948     syscall
949     la $a0, Vect_str
950     jal printStr
951     li $v0, 1
952     syscall
953     la $a0, Err_msg1
954     jal printStr
955     li $v0, 1
956     syscall
957     la $a0, Err_msg2
958     jal printStr
959     li $v0, 1
960     syscall
961     la $a0, Halt_msg
962     jal printStr
963     li $v0, 1
964     syscall
965     la $a0, Stop_msg
966     jal printStr
967     li $v0, 1
968     syscall
969     la $a0, end_data
970     jal printStr
971     li $v0, 1
972     syscall
973     .macro done
974         li $v0, 10 #stop code
975         syscall #stop
976     .end_macro
977     .macro print_mac (%str)
978         la $a0, %str
979         li $v0, 4
980         syscall
981     .end_macro
982     .macro msgbox (%str)
983         la $a0, %str
984         li $v0, 55 #GUI msg code
985         li $a1, 1 #msg type is info
986         syscall
987     .end_macro
988
989 ***ISR macro-> Trap
990 .macro _ISR (%str)
991     la $a0, %str
992     jal printStr
993     jal GUI_out
994     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
995     b loop_main
996 .end_macro
997 #code
998 .text
999 #saved:
1000 $t9
1001 li $t9, stop
1002 la $a0, header
1003 jal printStr
1004 #main Loop
1005 loop_main:
1006     subiu $t9, $t9, 1 #decrement counter
1007     blez $t9, Stop
1008     la $a0, prompt1
1009     jal printStr
1010     li $v0, 1
1011     syscall
1012     la $a0, prompt2
1013     jal printStr
1014     li $v0, 1
1015     syscall
1016     la $a0, Vect_str
1017     jal printStr
1018     li $v0, 1
1019     syscall
1020     la $a0, Err_msg1
1021     jal printStr
1022     li $v0, 1
1023     syscall
1024     la $a0, Err_msg2
1025     jal printStr
1026     li $v0, 1
1027     syscall
1028     la $a0, Halt_msg
1029     jal printStr
1030     li $v0, 1
1031     syscall
1032     la $a0, Stop_msg
1033     jal printStr
1034     li $v0, 1
1035     syscall
1036     la $a0, end_data
1037     jal printStr
1038     li $v0, 1
1039     syscall
1040     .macro done
1041         li $v0, 10 #stop code
1042         syscall #stop
1043     .end_macro
1044     .macro print_mac (%str)
1045         la $a0, %str
1046         li $v0, 4
1047         syscall
1048     .end_macro
1049     .macro msgbox (%str)
1050         la $a0, %str
1051         li $v0, 55 #GUI msg code
1052         li $a1, 1 #msg type is info
1053         syscall
1054     .end_macro
1055
1056 ***ISR macro-> Trap
1057 .macro _ISR (%str)
1058     la $a0, %str
1059     jal printStr
1060     jal GUI_out
1061     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
1062     b loop_main
1063 .end_macro
1064 #code
1065 .text
1066 #saved:
1067 $t9
1068 li $t9, stop
1069 la $a0, header
1070 jal printStr
1071 #main Loop
1072 loop_main:
1073     subiu $t9, $t9, 1 #decrement counter
1074     blez $t9, Stop
1075     la $a0, prompt1
1076     jal printStr
1077     li $v0, 1
1078     syscall
1079     la $a0, prompt2
1080     jal printStr
1081     li $v0, 1
1082     syscall
1083     la $a0, Vect_str
1084     jal printStr
1085     li $v0, 1
1086     syscall
1087     la $a0, Err_msg1
1088     jal printStr
1089     li $v0, 1
1090     syscall
1091     la $a0, Err_msg2
1092     jal printStr
1093     li $v0, 1
1094     syscall
1095     la $a0, Halt_msg
1096     jal printStr
1097     li $v0, 1
1098     syscall
1099     la $a0, Stop_msg
1100     jal printStr
1101     li $v0, 1
1102     syscall
1103     la $a0, end_data
1104     jal printStr
1105     li $v0, 1
1106     syscall
1107     .macro done
1108         li $v0, 10 #stop code
1109         syscall #stop
1110     .end_macro
1111     .macro print_mac (%str)
1112         la $a0, %str
1113         li $v0, 4
1114         syscall
1115     .end_macro
1116     .macro msgbox (%str)
1117         la $a0, %str
1118         li $v0, 55 #GUI msg code
1119         li $a1, 1 #msg type is info
1120         syscall
1121     .end_macro
1122
1123 ***ISR macro-> Trap
1124 .macro _ISR (%str)
1125     la $a0, %str
1126     jal printStr
1127     jal GUI_out
1128     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
1129     b loop_main
1130 .end_macro
1131 #code
1132 .text
1133 #saved:
1134 $t9
1135 li $t9, stop
1136 la $a0, header
1137 jal printStr
1138 #main Loop
1139 loop_main:
1140     subiu $t9, $t9, 1 #decrement counter
1141     blez $t9, Stop
1142     la $a0, prompt1
1143     jal printStr
1144     li $v0, 1
1145     syscall
1146     la $a0, prompt2
1147     jal printStr
1148     li $v0, 1
1149     syscall
1150     la $a0, Vect_str
1151     jal printStr
1152     li $v0, 1
1153     syscall
1154     la $a0, Err_msg1
1155     jal printStr
1156     li $v0, 1
1157     syscall
1158     la $a0, Err_msg2
1159     jal printStr
1160     li $v0, 1
1161     syscall
1162     la $a0, Halt_msg
1163     jal printStr
1164     li $v0, 1
1165     syscall
1166     la $a0, Stop_msg
1167     jal printStr
1168     li $v0, 1
1169     syscall
1170     la $a0, end_data
1171     jal printStr
1172     li $v0, 1
1173     syscall
1174     .macro done
1175         li $v0, 10 #stop code
1176         syscall #stop
1177     .end_macro
1178     .macro print_mac (%str)
1179         la $a0, %str
1180         li $v0, 4
1181         syscall
1182     .end_macro
1183     .macro msgbox (%str)
1184         la $a0, %str
1185         li $v0, 55 #GUI msg code
1186         li $a1, 1 #msg type is info
1187         syscall
1188     .end_macro
1189
1190 ***ISR macro-> Trap
1191 .macro _ISR (%str)
1192     la $a0, %str
1193     jal printStr
1194     jal GUI_out
1195     leq $0, $0 #Trap: simulate INT<-1 (in ktext)
1196     b loop_main
1197 .end_macro
1198 #code
1199 .text
1200 #saved:
1201 $t9
1202 li $t9, stop
1203 la $a0, header
1204 jal printStr
1205 #main Loop
1206 loop_main:
1207     subiu $t9, $t9, 1 #decrement counter
1208     blez $t9, Stop
1209     la $a0, prompt1
1210     jal print
```

```

Edit Execute
Lab4.asm
69 bles $t9, Stop
70 la $a0, prompt1
71 jal GUI_in #get Type in $a0
72 move $a0, $a0
73 #Int TYPE Branch table (if-case)
74 beq $a0, 0, NMI
75 beq $a0, 1, NVI
76 beq $a0, 2, VI
77 beq $a0, 3, Halt
78 b Err #none of above
79 NMI: _ISR(NMI_str)
80
81 NVI: _ISR(NVI_str)
82 VI: #get vector
83     la $a0, prompt2
84     jal GUI_in #get vector in $a0
85     move $t1, $a0 #save vector in $t1
86     _ISR(VI_str)
87 Halt: #Quit
88     la $a0, Halt_msg
89     jal GUI_out
90     jal printStr
91     done #**exit program**
92 Err: la $a0, Err_msg1 #default
93     jal GUI_out
94     b loop_main
95 Stop: la $a0, Stop_msg
96     jal printStr
97     done #**alt exit-stopped out
98 #end table
99 #---END MAIN LOOP---
100 #subroutines follow
101 #print $a0 on console
102 printStr:
103     li $v0, 4
104     syscall
105
Line: 158 Column: 59 [x] Show Line Numbers

```

```

Edit Execute
Lab4.asm
103     li $v0, 4
104     syscall
105     jr $ra
106 #OUTPUT GUI MSG
107 GUI_out: #ptr in $a0
108     li $v0, 55 #GUI msg code
109     li $a1, 1 #msg type is info
110     syscall
111     jr $ra
112 #INPUT GUI MSG
113 GUI_in: #a0=int, $a1=status code
114     li $v0, 51 #int read
115     syscall
116     bltz, $a1, in_error
117     jr $ra
118 in_error:
119     msgbox(Err_msg2)
120     li $a0, 5
121     jr $ra
122 #---end sub---
123 #**start handler code in kernel seg**
124 .macro push_k
125     move $k0, $a0 #save registers
126     move $k1, $a1
127 .end_macro
128 .macro pop_k
129     move $a0, $k0 #restore registers
130     move $a1, $k1
131 .end_macro
132 .macro _print %str
133     la $a0, %str
134     jal print_str
135     b return
136 .end_macro
137 #Begin kernel code
138 .text
139
Line: 158 Column: 59 [x] Show Line Numbers

```

```

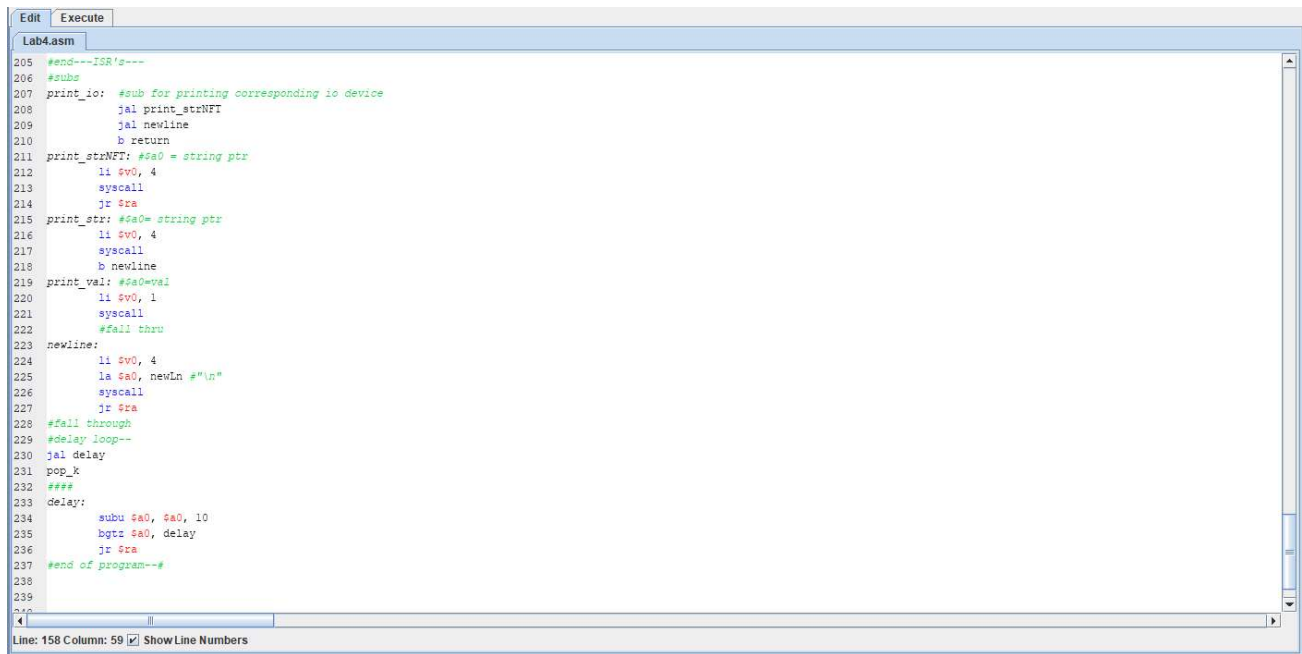
Edit Execute
Lab4.asm
137 #Begin Kernel code
138 .kdata
139 kmsg: .ascii "starting Interrupt handler for: "
140 def_msg: .ascii "error: unimplemented vector\n"
141 .align 2
142 newLn: .ascii "\n"
143 .align 2
144 end_kdata: .ascii "ENDKDATA$$$"
145 ioDevice1: .ascii " Keyboard"
146 ioDevice2: .ascii " Mouse"
147 ioDevice3: .ascii " Phone"
148 ioDevice4: .ascii " Speaker"
149 .ktext exc_seg
150 #save state
151 push_k
152 li $v0, 4
153 syscall
154 print_mac kmsg $prt_msg via macro
155 mfc0 $t0, $14 $EPC
156 addi $t0, $t0, 4 #incrementing RA in EPC
157 mco0 $t0, $14 $EPC+4 (for ERET)
158 #save: $t0=print code, $t1=vector: $$$=pointer to ioDevices
159 #---INT Branch Table---
160 Beq $t0, 0, NMI_Handler
161 Beq $t0, 1, NVI_Handler
162 #else VI
163 #---Branch Table---
164 Beq $t1, 0, v0
165 Beq $t1, 1, v1
166 Beq $t1, 2, v2
167 Beq $t1, 3, v3
168 #default
169 b def
170 #end Br table
171 #start ISR's
172 NMI_Handler:
Line: 158 Column: 59 Show Line Numbers

```

```

Edit Execute
Lab4.asm
171 #start ISR's
172 NMI_Handler:
173 _print NMI_str
174 NVI_Handler:
175 _print NVI_str
176 #---Vector Table---
177 v0:
178     la $s5, ioDevice1
179     b finish#ioDevice1
180 v1:
181     la $s5, ioDevice2
182     b finish#ioDevice2
183 v2:
184     la $s5, ioDevice3
185     b finish#ioDevice3
186 v3:
187     la $s5, ioDevice4
188     b finish#ioDevice4
189 def: #un-impl
190     print_mac def_msg
191     eret
192 finish:
193     print_mac Vect_str
194     move $a0, $t1 #set vector
195     jal print_val
196     la $a0, ($s5) #set IO device
197     jal print_io
198     b return
199 #end Tabledef: #un-impl
200 print_mac def_msg
201 return:
202     pop_k
203     eret
204 #end---ISR's---
205 #unimpl
Line: 158 Column: 59 Show Line Numbers

```



```
Lab4.asm
205 #end--ISR's--
206 #subs
207 print_io: #sub for printing corresponding io device
208     jal print_strNFI
209     jal newline
210     b return
211 print_strNFI: #a0 = string ptr
212     li v0, 4
213     syscall
214     jr $ra
215 print_str: #a0= string ptr
216     li v0, 4
217     syscall
218     b newline
219 print_val: #a0=val
220     li v0, 1
221     syscall
222     #fall thru
223 newline:
224     li v0, 4
225     la $a0, newLn #"\\n"
226     syscall
227     jr $ra
228 #fall through
229 #delay loop--
230 jal delay
231 pop_k
232 ###
233 delay:
234     subu $a0, $a0, 10
235     bgtz $a0, delay
236     jr $ra
237 #end of program--#
238
239
240
```

Line: 158 Column: 59 ☒ Show Line Numbers

(END of Source Code)