

QtSPin

FileSimulatorRegistersText SegmentData SegmentWindowHelp

FP RegsInt Regs [16]

DataText

Int Regs [16]

HI = 0
LO = 0

R0 [r0] = 0
R1 [at] = 1
R2 [v0] = b
R3 [v1] = 0
R4 [a0] = 53
R5 [a1] = 7ffff754
R6 [a2] = 7ffff760
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 48
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0
R28 [gp] = 10008000
R29 [sp] = 7ffff700
R30 [s8] = 0
R31 [ra] = 400018

Text

User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw \$4, 0(\$29) ; 183: lw \$a0 0(\$sp) # argv
[00400004] 27a50004 addiu \$5, \$29, 4 ; 184: addiu \$a1 \$sp 4 # argv
[00400008] 24a60004 addiu \$6, \$5, 4 ; 185: addiu \$a2 \$a1 4 # envp
[0040000c] 00041080 sll \$2, \$4, 2 ; 186: sll \$v0 \$a0 2
[00400010] 00c23021 addu \$6, \$6, \$2 ; 187: addu \$a2 \$a2 \$v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400024] 34090000 ori \$9, \$0, 0 ; 10: li \$t1, 0 #To keep track of how many characters on stack
[00400028] 3c01ffff lui \$1, -1 ; 11: li \$t2, -8 #For stack pointer math later
[0040002c] 342afff8 ori \$10, \$1, -8
[00400030] 3402000c ori \$2, \$0, 12 ; 14: li \$v0, 12 #Set \$v0 to 12 to read in character
[00400034] 21290001 addi \$9, \$9, 1 ; 16: addi \$t1, \$t1, 1 #A character is being put onto the stack
[00400038] 214a0004 addi \$10, \$10, 4 ; 17: addi \$t2, \$t2, 4 #For stack pointer math later
[0040003c] 23bdffff addi \$29, \$29, -4 ; 18: addi \$sp, \$sp, -4 #Moving
[00400040] 0000000c syscall ; 20: syscall #Reading in charac
[00400044] afa20000 sw \$2, 0(\$29) ; 22: sw \$v0, (\$sp) #Storing read
[00400048] 3401003f ori \$1, \$0, 63 ; 24: bne \$v0, 63, readIn #Branch
[0040004c] 1422ffff bne \$1, \$2, -28 [readIn-0x0040004c]
[00400050] 34020004 ori \$2, \$0, 4 ; 27: li \$v0, 4 #syscall for str
[00400054] 3c041001 lui \$4, 4097 [msg1] ; 28: la \$a0, msg1 #Loading "\n"
[00400058] 0000000c syscall ; 29: syscall #Printing
[0040005c] 23bd0004 addi \$29, \$29, 4 ; 31: addi \$sp, \$sp, 4 #Moving s
[00400060] 2129ffff addi \$9, \$9, -1 ; 32: addi \$t1, \$t1, -1 #'?' rem
[00400064] 03aae821 addu \$29, \$29, \$10 ; 34: addu \$sp, \$sp, \$t2 #\$sp =
[00400068] 3402000b ori \$2, \$0, 11 ; 36: li \$v0, 11 #Set \$v0 to 11
[0040006c] 34040000 ori \$4, \$0, 0 ; 39: li \$a0, 0 #Setting \$a0 to
[00400070] 8fa40000 lw \$4, 0(\$29) ; 41: lw \$a0, (\$sp) #Moving char
[00400074] 23bdffff addi \$29, \$29, -4 ; 42: addi \$sp, \$sp, -4 #Moving
[00400078] 2129ffff addi \$9, \$9, -1 ; 43: addi \$t1, \$t1, -1 #A chara
[0040007c] 2888005b slti \$8, \$4, 91 ; 46: slti \$t0, \$a0, 91 #If char
[00400080] 34010001 ori \$1, \$0, 1 ; 47: beq \$t0, 1, toLower #Branch
[00400084] 10280003 beq \$1, \$8, 12 [toLower-0x00400084]

Console

aBCdeFGhijKLMNOpqRs?
AbcDEfgHIJklmnoPqRs]

Memory and registers cleared

SPIM Version 9.1.21 of January 17, 2020
Copyright 1990-2017 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
Attempt to execute non-instruction at 0x0040009c