```
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)            ; 183: lw $a0 0($sp) # argc
[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] 3c011001  lui $1, 4097 [hdr]       ; 70: la $a0, hdr
[00400028] 3424026c  ori $4, $1, 620 [hdr]
[0040002c] 34020004  ori $2, $0, 4            ; 71: li $v0, 4
[00400030] 0000000c  syscall                  ; 72: syscall # print header
[00400034] 3408002a  ori $8, $0, 42           ; 78: li $t0, 42
[00400038] 3c010001  lui $1, 1                ; 79: li $t1, 65539
[0040003c] 34290003  ori $9, $1, 3
[00400040] 3c01ffff  lui $1, -1               ; 80: li $t2, -42
[00400044] 342affd6  ori $10, $1, -42
[00400048] 3c01fffe  lui $1, -2               ; 81: li $t3, -65539
[0040004c] 342bfffd  ori $11, $1, -3
[00400050] 3c010001  lui $1, 1                ; 83: add $t0, $t0, 100000
[00400054] 342186a0  ori $1, $1, -31072
[00400058] 01014020  add $8, $8, $1
[0040005c] 0109082a  slt $1, $8, $9           ; 85: blt $t0, $t1, testLabel
[00400060] 1420000b  bne $1, $0, 44 [testLabel-0x00400060]
[00400064] 0128082a  slt $1, $9, $8           ; 86: ble $t0, $t1, testLabel
[00400068] 10200009  beq $1, $0, 36 [testLabel-0x00400068]
[0040006c] 0128082a  slt $1, $9, $8           ; 87: bgt $t0, $t1, testLabel
[00400070] 14200007  bne $1, $0, 28 [testLabel-0x00400070]
[00400074] 0109082a  slt $1, $8, $9           ; 88: bge $t0, $t1, testLabel
[00400078] 10200005  beq $1, $0, 20 [testLabel-0x00400078]
[0040007c] 000840c0  sll $8, $8, 3            ; 90: sll $t0, $t0, 3
[00400080] 01284004  sllv $8, $8, $9          ; 91: sll $t0, $t0, $t1
[00400084] 00954020  add $8, $4, $21          ; 93: add $t0, $a0, $s5
[00400088] 08100023  j 0x0040008c [testLabel] ; 96: j testLabel # useless
[0040008c] 3c081001  lui $8, 4097 [list]      ; 102: la $t0, list # set $t0
addr of the array
[00400090] 3c011001  lui $1, 4097             ; 103: lw $t1, len # set $t1 to
length
[00400094] 8c290258  lw $9, 600($1)
[00400098] 8d0a0000  lw $10, 0($8)            ; 105: lw $t2, ($t0) # set min,
$t2 to array[0]
[0040009c] 8d0b0000  lw $11, 0($8)            ; 106: lw $t3, ($t0) # set max,
$t3 to array[0]
[004000a0] 34100000  ori $16, $0, 0           ; 107: li $s0, 0 # set sum=0
[004000a4] 340f0000  ori $15, $0, 0           ; 108: li $t7, 0 # set count of
odd values=0
[004000a8] 34180000  ori $24, $0, 0           ; 109: li $t8, 0 # set avg=0
[004000ac] 8d0d0000  lw $13, 0($8)            ; 111: lw $t5, ($t0) # get
list[n]
[004000b0] 34010002  ori $1, $0, 2            ; 113: rem $t6, $t5, 2 # store
remainder in $t6
[004000b4] 01a1001a  div $13, $1
[004000b8] 00007010  mfhi $14
[004000bc] 11c00003  beq $14, $0, 12 [skipAdd-0x004000bc]
[004000c0] 21ef0001  addi $15, $15, 1         ; 116: add $t7, $t7, 1 #
increment number of odd values
[004000c4] 020d8020  add $16, $16, $13        ; 118: add $s0, $s0, $t5 # sum
= sum+list[n]
[004000c8] 11c00004  beq $14, $0, 16 [notMin-0x004000c8]
[004000cc] 01aa082a  slt $1, $13, $10         ; 124: bge $t5, $t2, notMin #
is new min?
[004000d0] 10200002  beq $1, $0, 8 [notMin-0x004000d0]
[004000d4] 000d5021  addu $10, $0, $13        ; 125: move $t2, $t5 # set new
min (into $t2)
[004000d8] 11c00004  beq $14, $0, 16 [notMax-0x004000d8]
[004000dc] 016d082a  slt $1, $11, $13         ; 130: ble $t5, $t3, notMax #
is new max?
[004000e0] 10200002  beq $1, $0, 8 [notMax-0x004000e0]
[004000e4] 000d5821  addu $11, $0, $13        ; 131: move $t3, $t5 # set new
max (into $t5)
[004000e8] 2129ffff  addi $9, $9, -1          ; 133: sub $t1, $t1, 1 #
decrement length counter
[004000ec] 21080004  addi $8, $8, 4           ; 134: add $t0, $t0, 4 #
increment addr by word (+4 bytes)
[004000f0] 1520ffef  bne $9, $0, -68 [loop-0x004000f0]
[004000f4] 15e00002  bne $15, $0, 8           ; 137: div $t8, $s0, $t7 # avg
= sum / count (odd)
[004000f8] 0000000d  break
[004000fc] 020f001a  div $16, $15
[00400100] 0000c012  mflo $24
[00400104] 3c011001  lui $1, 4097             ; 139: sw $t2, min # save min
[00400108] ac2a025c  sw $10, 604($1)
[0040010c] 3c011001  lui $1, 4097             ; 140: sw $t3, max # save max
```

```
[00400110] ac2b0260  sw $11, 608($1)
[00400114] 3c011001  lui $1, 4097             ; 141: sw $s0, sum # save sum
[00400118] ac300264  sw $16, 612($1)
[0040011c] 3c011001  lui $1, 4097             ; 142: sw $t8, avg # save avg
[00400120] ac380268  sw $24, 616($1)
[00400124] 3c011001  lui $1, 4097 [a1Msg]     ; 150: la $a0, a1Msg
[00400128] 34240291  ori $4, $1, 657 [a1Msg]
[0040012c] 34020004  ori $2, $0, 4            ; 151: li $v0, 4
[00400130] 0000000c  syscall                  ; 152: syscall # print "min = "
[00400134] 3c011001  lui $1, 4097             ; 154: lw $a0, min
[00400138] 8c24025c  lw $4, 604($1)
[0040013c] 34020001  ori $2, $0, 1            ; 155: li $v0, 1
[00400140] 0000000c  syscall                  ; 156: syscall # print min
[00400144] 3c011001  lui $1, 4097 [a2Msg]     ; 158: la $a0, a2Msg
[00400148] 3424029e  ori $4, $1, 670 [a2Msg]
[0040014c] 34020004  ori $2, $0, 4            ; 159: li $v0, 4
[00400150] 0000000c  syscall                  ; 160: syscall # print "max = "
[00400154] 3c011001  lui $1, 4097             ; 162: lw $a0, max
[00400158] 8c240260  lw $4, 608($1)
[0040015c] 34020001  ori $2, $0, 1            ; 163: li $v0, 1
[00400160] 0000000c  syscall                  ; 164: syscall # print max
[00400164] 3c011001  lui $1, 4097 [a3Msg]     ; 166: la $a0, a3Msg
[00400168] 342402ac  ori $4, $1, 684 [a3Msg]
[0040016c] 34020004  ori $2, $0, 4            ; 167: li $v0, 4
[00400170] 0000000c  syscall                  ; 168: syscall # print "sum = "
[00400174] 3c011001  lui $1, 4097             ; 170: lw $a0, sum
[00400178] 8c240264  lw $4, 612($1)
[0040017c] 34020001  ori $2, $0, 1            ; 171: li $v0, 1
[00400180] 0000000c  syscall                  ; 172: syscall # print sum
[00400184] 3c011001  lui $1, 4097 [a4Msg]     ; 174: la $a0, a4Msg
[00400188] 342402ba  ori $4, $1, 698 [a4Msg]
[0040018c] 34020004  ori $2, $0, 4            ; 175: li $v0, 4
[00400190] 0000000c  syscall                  ; 176: syscall
[00400194] 3c011001  lui $1, 4097             ; 178: lw $a0, avg
[00400198] 8c240268  lw $4, 616($1)
[0040019c] 34020001  ori $2, $0, 1            ; 179: li $v0, 1
[004001a0] 0000000c  syscall                  ; 180: syscall
[004001a4] 3c011001  lui $1, 4097 [new_ln]    ; 182: la $a0, new_ln # print a
newline
[004001a8] 3424028f  ori $4, $1, 655 [new_ln]
[004001ac] 34020004  ori $2, $0, 4            ; 183: li $v0, 4
[004001b0] 0000000c  syscall                  ; 184: syscall
[004001b4] 3402000a  ori $2, $0, 10           ; 189: li $v0, 10
```

```
[004001b8] 0000000c  syscall                  ; 190: syscall # all done!

Kernel Text Segment [80000000]..[80010000]
[80000180] 0001d821  addu $27, $0, $1         ; 90: move $k1 $at # Save $at
[80000184] 3c019000  lui $1, -28672           ; 92: sw $v0 s1 # Not
re-entrant and we can't trust $sp
[80000188] ac220200  sw $2, 512($1)
[8000018c] 3c019000  lui $1, -28672           ; 93: sw $a0 s2 # But we need
to use these registers
[80000190] ac240204  sw $4, 516($1)
[80000194] 401a6800  mfc0 $26, $13            ; 95: mfc0 $k0 $13 # Cause
register
[80000198] 001a2082  srl $4, $26, 2           ; 96: srl $a0 $k0 2 # Extract
ExcCode Field
[8000019c] 3084001f  andi $4, $4, 31          ; 97: andi $a0 $a0 0x1f
[800001a0] 34020004  ori $2, $0, 4            ; 101: li $v0 4 # syscall 4
(print_str)
[800001a4] 3c049000  lui $4, -28672 [__m1_]   ; 102: la $a0 __m1_
[800001a8] 0000000c  syscall                  ; 103: syscall
[800001ac] 34020001  ori $2, $0, 1            ; 105: li $v0 1 # syscall 1
(print_int)
[800001b0] 001a2082  srl $4, $26, 2           ; 106: srl $a0 $k0 2 # Extract
ExcCode Field
[800001b4] 3084001f  andi $4, $4, 31          ; 107: andi $a0 $a0 0x1f
[800001b8] 0000000c  syscall                  ; 108: syscall
[800001bc] 34020004  ori $2, $0, 4            ; 110: li $v0 4 # syscall 4
(print_str)
[800001c0] 3344003c  andi $4, $26, 60         ; 111: andi $a0 $k0 0x3c
[800001c4] 3c019000  lui $1, -28672           ; 112: lw $a0 __excp($a0)
[800001c8] 00240821  addu $1, $1, $4
[800001cc] 8c240180  lw $4, 384($1)
[800001d0] 00000000  nop                      ; 113: nop
[800001d4] 0000000c  syscall                  ; 114: syscall
[800001d8] 34010018  ori $1, $0, 24           ; 116: bne $k0 0x18 ok_pc # Bad
PC exception requires special checks
[800001dc] 143a0008  bne $1, $26, 32 [ok_pc-0x800001dc]
[800001e0] 00000000  nop                      ; 117: nop
[800001e4] 40047000  mfc0 $4, $14             ; 119: mfc0 $a0 $14 # EPC
[800001e8] 30840003  andi $4, $4, 3           ; 120: andi $a0 $a0 0x3 # Is
EPC word-aligned?
[800001ec] 10040004  beq $0, $4, 16 [ok_pc-0x800001ec]
[800001f0] 00000000  nop                      ; 122: nop
[800001f4] 3402000a  ori $2, $0, 10           ; 124: li $v0 10 # Exit on
```

```
really bad PC
[800001f8] 0000000c  syscall                    ; 125: syscall
[800001fc] 34020004  ori $2, $0, 4              ; 128: li $v0 4 # syscall 4
(print_str)
[80000200] 3c019000  lui $1, -28672 [__m2_]     ; 129: la $a0 __m2_
[80000204] 3424000d  ori $4, $1, 13 [__m2_]
[80000208] 0000000c  syscall                    ; 130: syscall
[8000020c] 001a2082  srl $4, $26, 2             ; 132: srl $a0 $k0 2 # Extract
ExcCode Field
[80000210] 3084001f  andi $4, $4, 31            ; 133: andi $a0 $a0 0x1f
[80000214] 14040002  bne $0, $4, 8 [ret-0x80000214]; 134: bne $a0 0 ret # 0
means exception was an interrupt
[80000218] 00000000  nop                        ; 135: nop
[8000021c] 401a7000  mfc0 $26, $14              ; 145: mfc0 $k0 $14 # Bump EPC
register
[80000220] 275a0004  addiu $26, $26, 4          ; 146: addiu $k0 $k0 4 # Skip
faulting instruction
[80000224] 409a7000  mtc0 $26, $14              ; 148: mtc0 $k0 $14
[80000228] 3c019000  lui $1, -28672             ; 153: lw $v0 s1 # Restore
other registers
[8000022c] 8c220200  lw $2, 512($1)
[80000230] 3c019000  lui $1, -28672             ; 154: lw $a0 s2
[80000234] 8c240204  lw $4, 516($1)
[80000238] 001b0821  addu $1, $0, $27           ; 157: move $at $k1 # Restore
$at
[8000023c] 40806800  mtc0 $0, $13               ; 160: mtc0 $0 $13 # Clear
Cause register
[80000240] 401a6000  mfc0 $26, $12              ; 162: mfc0 $k0 $12 # Set
Status register
[80000244] 375a0001  ori $26, $26, 1            ; 163: ori $k0 0x1 # Interrupts
enabled
[80000248] 409a6000  mtc0 $26, $12              ; 164: mtc0 $k0 $12
[8000024c] 42000018  eret                       ; 167: eret
```

Example program to find max/min

```
min (odd) = 51
max (odd) = 999
sum (odd) = 21901
avg (odd) = 300
```