

# Lab 3

Andrew Chang-DeWitt

## Task 1

Screenshot of run w/  $A=(7,12,3,6,23,90,-2,-122,10,1)$  :

The screenshot displays the Chocopy simulator interface. The main window shows assembly code with columns for PC, Machine Code, Basic Code, and Original Code. The code is a MIPS-like assembly for finding the maximum value in an array. The registers window on the right shows the current state of registers x0 through x31. The output window at the bottom shows the array elements: 7, 12, 3, 6, 23, -2, -122, 10, 1, 90.

PC	Machine Code	Basic Code	Original Code
0x0	0x10000417	swipc x8 65536	main: la x0 A # &A <- pointer to A[0]
0x4	0x00000413	addi x8 x0 0	main: la x0 A # &A <- pointer to A[0]
0x8	0x02440493	addi x9 x8 36	addi x1 x0 36 # ptr to A[n-1] to know when to stop processing A
0xc	0x00002905	lw x18 0(x8)	lw s2 0(x0) # max value so far <- a_i
0x10	0x00004903	add x19 x8 x0	add s3 s0 x0 # addr of max so far <- addr a_i
0x14	0x00004013	addi x20 x8 4	addi s4 s0 4 # i <- temp value track current loop iteration (1..n-1),
0x18	0x00002303	lw x6 0(x20)	lw t1 0(s4) # get a_i
0x1c	0x01234663	blt x6 x18 12	blt t1 s2 endif # if a_i < max, do nothing
0x20	0x00003093	add x18 x6 x0	add s2 t1 x0 # else update max to a_i
0x24	0x00004903	add x19 x20 x0	add s3 s4 x0 # and max index to addr of a_i
0x28	0x00004013	addi x20 x20 4	endif: addi s4 s4 4 # i++
0x2c	0x0f9a46e3	blt x20 x9 -20	blt s4 s1 loop # loop again if i still < n-1
0x30	0x000043a3	lw x6 4(x19)	lw t1 4(s3) # get value 1 past current addr
0x34	0x00004023	sw x6 0(x19)	sw t1 0(s3) # and store it in current address

Registers Memory Cache VDB

Address	x3	x2	x1	x0
0x10000030	0	0	0	0
0x1000002C	0	0	0	0
0x10000028	0	0	32	44
0x10000024	0	0	0	90
0x10000020	0	0	0	1
0x1000001C	0	0	0	90
0x10000018	-1	-1	-1	-122
0x10000014	-1	-1	-1	-2
0x10000010	0	0	0	23
0x1000000C	0	0	0	6
0x10000008	0	0	0	3
0x10000004	0	0	0	12
0x10000000	0	0	0	7

Jump to: -- choose -- Up Down

Address: Go

Copy! Download! Clear!

7, 12, 3, 6, 23, -2, -122, 10, 1, 90

Display Settings Decimal

Screenshot of run w/  $A=(1205,5523,703,66,-324,0,-9,80,5048,990)$  :

Venus Editor Simulator Chocopy

Run Stop Prev Reset Dump Trace Re-assemble from Editor

PC	Machine Code	Basic Code	Original Code
0x0	0x10000417	auipc x8 65536	main: la s0 A # &A <- pointer to A[0]
0x4	0x00000413	addi x8 x8 0	main: la s0 A # &A <- pointer to A[0]
0x8	0x02440493	addi x9 x8 36	addi t1 s0 36 # ptr to A[n-1] to know when to stop processing A
0xc	0x00042903	lw x18 0(x8)	lw s2 0(s0) # max value so far <- a_1
0x10	0x00040953	add x19 x8 x0	add s2 s0 x8 # add of max so far <- addr a_1
0x14	0x00040413	addi x20 x8 4	addi s4 s0 4 # i <- temp value track current loop iteration (1..n-1),
0x18	0x00042903	lw x6 0(x20)	lw t1 0(s4) # get a_i
0x1c	0x01234663	blt x6 x18 12	blt t1 s2 endif # if a_i < max, do nothing
0x20	0x00030933	add x18 x6 x0	add s2 t1 x8 # else update max to a_i
0x24	0x00040953	add x19 x20 x0	add s3 s4 x8 # and max index to addr-of a_1
0x28	0x00040413	addi x20 x20 4	endif: addi s4 s4 4 # i++
0x2c	0x0f546663	blt x20 x9 -20	blt s4 s1 loop # loop again if i still < n-1
0x30	0x00494303	lw x6 4(x19)	lw t1 4(s3) # get value i past current addr
0x34	0x00050023	sw x6 0(x19)	sw t1 0(s3) # and store it in current code

Copy! Download! Clear!

1705, 703, 66, -324, 0, -9, 80, 5048, 990, 5523

Registers Memory Cache VDB

Address	x3	x2	x1	x0
0x10000030	0	0	0	0
0x1000003C	0	0	0	0
0x10000028	0	0	32	44
0x10000024	0	0	21	-100
0x10000020	0	0	3	-34
0x1000001C	0	0	19	-72
0x10000018	0	0	0	100
0x10000014	-1	-1	-1	-9
0x10000010	0	0	0	0
0x1000000C	-1	-1	-2	-68
0x10000008	0	0	0	166
0x10000004	0	0	2	-65
0x10000000	0	0	4	-15

Jump to --choose-- Up Down

Address: Go

Display Settings

## Task 2

Screenshot of run w/ A=(10,20,30,40), B=(90,80,70,60,50) :

Venus Editor Simulator Chocopy Decoder Tester

Run Stop Prev Reset Dump Trace Re-assemble from Editor

PC	Machine Code	Basic Code	Original Code
0x0	0x10000517	auipc x10 65536	la a0 A # &A <- pointer to A[0]
0x4	0x00050513	addi x10 x10 0	la a0 A # &A <- pointer to A[0]
0x8	0x10000297	auipc x5 65536	la t0 x # &x
0xc	0x01C28293	addi x5 x5 28	la t0 x # &x
0x10	0x0002A583	lw x11 0(x5)	lw a1 0(t0) # x <- *t0
0x14	0x10000617	auipc x12 65536	la a2 B # &B <- pointer to B[0]
0x18	0xFFC60613	addi x12 x12 -4	la a2 B # &B <- pointer to B[0]
0x1c	0x10000297	auipc x5 65536	la t0 y # &y
0x20	0x00C28293	addi x5 x5 12	la t0 y # &y
0x24	0x0002A683	lw x13 0(x5)	lw a3 0(t0) # y <- *t0
0x28	0x00C000EF	jal x1 12	jal ra sum_arr
0x2c	0x08C000EF	jal x1 188	jal ra print_arr
0x30	0x10C0006F	jal x0 268	j exit

Copy! Download! Clear!

100, 100, 100, 100, 50  
Found 0 warnings!

Registers Memory Cache VDB

Integer (R) Floating (F)

zero 0x00000000

ra (x1) 0x00000030

sp (x2) 0x7FFFFFFC

gp (x3) 0x10000000

tp (x4) 0x00000000

t0 (x5) 0x10000030

t1 (x6) 0x1000003C

t2 (x7) 0x10000040

s0 (x8) 0x10000014

s1 (x9) 0x10000010

a0 (x10) 0x00000004

a1 (x11) 0x10000006

a2 (x12) 0x10000010

a3 0x00000014

Display Settings Hex

Screenshot of run w/ A=(3,2,1,0,1,2,3), B=(7,8,9,10,9,8,7) :

https://venus.cs61c.org

VenusEditorSimulatorChocopyDecoderTester

RunStepPrevResetDumpTraceRe-assemble from Editor

0x0	0x10000517	auipc x10 65536	la a0 A # 8A <- pointer to A[0]
0x4	0x00050513	addi x10 x10 0	la a0 A # 8A <- pointer to A[0]
0x8	0x10000297	auipc x5 65536	la t0 x # 8x
0xc	0x03028293	addi x5 x5 48	la t0 x # 8x
0x10	0x0002A583	lw x11 0(x5)	lw a1 0(t0) # x <- *t0
0x14	0x10000617	auipc x12 65536	la a2 B # 8B <- pointer to B[0]
0x18	0x00060613	addi x12 x12 8	la a2 B # 8B <- pointer to B[0]
0x1c	0x10000297	auipc x5 65536	la t0 y # 8y
0x20	0x02028293	addi x5 x5 32	la t0 y # 8y
0x24	0x0002A683	lw x13 0(x5)	lw a3 0(t0) # y <- *t0
0x28	0x00C000EF	jal x1 12	jal ra sum_arr
0x2c	0x00C000EF	jal x1 188	jal ra print_arr
0x30	0x10C0006F	jal x0 268	j exit
0x34 0xffff0000 addi x0 x0 0 addi x0 x0 0 addi x0 x0 0 addi x0 x0 0			

Copy!Download!Clear

10, 10, 10, 10, 10, 10  
Found 0 warnings!

RegistersMemoryCacheVDB

Integer (R)Floating (F)

zero0x00000000

ra (x1)0x00000010

sp (x2)0xffffffff

gp (x3)0x10000000

tp (x4)0x00000000

t0 (x5)0x1000000C

t1 (x6)0x10000010

t2 (x7)0x1000000C

s0 (x8)0x10000010

s1 (x9)0x1000001C

a0(x10)0x00000000

a1(x11)0x10000002

a2(x12)0x1000001C

a3...

Display SettingsMax

Exited with error code 0