ECE 484/584

Tutorial on Using QtSpim

QtSpim is software that will help you to simulate the execution of MIPS assembly programs. It does a context and syntax check while loading an assembly program. In addition, it adds in necessary overhead instructions as needed, and updates register and memory content as each instruction is executed. Below, is a tutorial on how to use QtSpim.

Where to Get QtSpim?

Download the source from the SourceForge.org link at: http://pages.cs.wisc.edu/~larus/spim.html (See "New versions of spim" text in red at the top of the page.) Alternatively, you can go directly to: http://sourceforge.net/projects/spimsimulator/files/ - Note that versions for Windows machines, Linux machines, and Macs are all available

Important Documents to Read :

Kindly make a point to read the below documents before starting, from the Appendix A of the third edition of Hennessy & Patterson, *Computer Organization and Design: The Hardware/Software Interface*. This documentation is far more complete and up-to-date than the documentation included in the *spim* distribution. :

Assemblers, Linkers, and the SPIM Simulator (PDF). An overview and reference manual for *spim* and the MIPS32 instruction set.

Getting Started with spim (<u>PDF</u>). Overview of the console version of *spim* (both Unix and Windows).

Getting Started with xspim (PDF). Overview of the X-windows version of spim.

Getting Starting with PCSpim (PDF). Overview of the Microsoft Windows version of spim.

SPIM Command-Line Options (PDF). Overview of the command line options of *spim* (all versions).

SPIM in action

When you open QtSpim, A window will open as shown in Figure 1. The window is divided into different sections:

- 1. The *Register* tabs display the content of all registers.
- 2. Buttons across the top are used to load and run a simulation

3. The *Text* tab displays the MIPS instructions loaded into memory to be executed. (From left-to-right, the memory address of an instruction, the contents of the address in hex, the actual MIPS instructions – where register numbers are used, the MIPS assembly that you wrote, and any comments you made in your code are displayed.)

4. The *Data* tab displays memory addresses and their values in the data and stack segments of the memory.

5. The *Information Console* lists the actions performed by the simulator.

e QtSpim File Simulator Registers Text Segment Data Segment W	indow Help			(WAR BOARD)	
2 2 9 9 3 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9					
FP Regs Int Regs [16]	Data	Text			
Regs [16]	🗗 🗙 Text				
= 400078	<u>^</u>			User Text Segment [00400000][00440000]	
2C = 0	[00400000] 8fa40000	1w \$4, D(\$29)	; 183: lv \$a0 0(\$sp) # argc	
dyAddr = 0	[00400004	1] 27a50004	addiu \$5, \$29, 4	; 184: addiu Sal Sap 4 # argv	
atus = 3000ff10	[00400000	1 00041080	s11 \$2, \$4, 2	; 100; audiu saz sai + + envp ; 186; sli Svo San 2	
	[00400010	01 00c23021	addu \$6, \$6, \$2	; 187; addu \$a2 \$a2 \$v0	
- 0	[00400014] 0c100009	jal 0x00400024 [main]	: 188: jal main	
= 0	[00400018	8] 00000000 [8	nop	; 189: nop	
[r0] = 0	[0040001c	3402000a	ori \$2, \$0, 10	; 191: li \$v0 10	
[at] = 10010000	[00400020	01 0000000c	syscall	; 192: syscall # syscall 10 (exit)	
[v0] = a	[00400024	<pre>3c011001 3c011001</pre>	lui \$1, 4097	; 24: 1v \$s0, N # load loop counter into \$s0	
[v1] = 0	[00400020	1 3c011001	lui \$1, 4097 [X]	: 26: la StO. X # load the address of X into StO	
[a0] = 1e	[00400030	34280004	ori \$8, \$1, 4 [X]		
[a1] = 7ffff628	[00400034	1 02208824	and \$17, \$17, \$0	; 28: and \$s1, \$s1, \$zero # clear \$s1 aka temp sum	
[a3] = 0	E [00400038	8] 8d090000	lw \$9, 0(\$8)	; 30: lv \$t1, 0(\$t0) # load the next value of x	
[t0] = 10010018	[0040003c	2] 02298820	add \$17, \$17, \$9	; 32: add \$s1, \$s1, \$t1 # add it to the running sum	
9 [t1] = a	[00400040	0] 21080004	addi \$8, \$8, 4	; 34: addi \$t0, \$t0, 4 # increment to the next address	
[0 [t2] = 0	[00400044	1] 2210ffff	addi \$16, \$16, -1	; 36: addi 580, 580, -1 # decrement the loop counter	
12 [t4] = 0	[00400046	3 30011001	Ini \$1, 4097	p-uxuuquuuqoj . 4.0. ey Sel. SIM ž etore the final total	
13 [t5] = 0	[00400050	ac310018	sw \$17, 24(\$1)	, to be obe, but y boot one what could	
14 [t6] = 0	[00400054	34020004	ori \$2, \$0, 4	; 43: li \$v0, 4 # system call code for print str	
15 [t7] = 0	[00400058	3c011001	lui \$1, 4097 [str]	: 45: la \$a0, str # address of string to print	
16 [s0] = 0	[0040005c	3424001c	ori \$4, \$1, 28 [str]		
17 [S1] = 1e	[00400060	0] 0000000c	syscall	; 47: syscall	
19 [s3] = 0	[00400064	1 34020001	or1 \$2, \$0, 1	; 50: 11 SVO, 1 # system call code for print str	
20 [84] = 0	[00400066	1 8c240018	1w \$4, 24(\$1)	; 52: IV 980, SUM # address of string to print	
21 [s5] = 0	[00400070	00000000c	syscall	: 54: syscall	
22 [s6] = 0	[00400074] 3402000a	ori \$2, \$0, 10	: 58: li \$v0, 10 # syscall to exit cleanly from main only	
23 [87] = 0	[00400078	1 000000c	syscall	; 60: syscall # this ends execution	
25 [t9] = 0					
26 [k0] = 0					
27 [k1] = 0	*				
yyright 1990-2010, James R. Larus. Rights Reserved. Mis distributed under a BSD license. e the file README for a full copyright notice.					
mory and registers cleared sket (: / Jonen ; https://min.jong/imen.jong/ more ; https://min.jong/ more ; https://min.jong/ Rights Reserved. Rights Reserved. Bights Reserved. the ResEACOME for a HISO location. e the ResEACOME for a HISO location.					
ded: C:/Users/student/AppData/Local/Temp/qt_temp.kd/s32 M Version 9.1.* of September 4, 2011 synph: 1990-2010, James R. Larus. Rights Reserved. M & datributed under a BSD license. the file README for a full copyright notice.					
9 🙆 🖸 🚞 🧿 🜌	I	W			▲ 🍽 🐑 🚯 4:45 PM

Figure 1 : QtSpim

To run the program in QtSpim:

1. Use a text editor to create your program yyyyyy.s

2. Click on the "load" button and open yyyyy.s

3. You can then run the program by simply pressing the "run" (play) button – all instructions will be executed, and the final contents of memory and the register file will be reflected in the QtSpim window.

Debugging

Suppose your program does not do what you expect. What can you do? QtSpim has two features that help debug your program.

The first, and perhaps the most useful, is single-stepping, which allows you to run your program an instruction at a time. The single stepping icon can be found in the toolbar. Every time you do single stepping, QtSpim will execute one instruction and update its display, so that you can see what the instruction changed in the registers or memory.

What do you do if your program runs for a long time before the bug arises? You could singlestep until you get to the bug, but that can take a long time. A better alternative is to use a *breakpoint*, which tells QtSpim to stop your program immediately before it executes a particular instruction. When QtSpim is about to execute the instruction where there is a breakpoint, it asks for continue, single stepping or abort.

Single-stepping and setting breakpoints will probably help you find a bug in your program quickly. How do you fix it? Go back to the editor that you used to create your program and change it. Click on the Riinitialize simulator tab in the toolsbar and load the sourcefile again.

Generally Useful Information

When using QtSpim, you may find the following information to be useful:

You can access all of the commands via the "File" and "Simulator" menus as well.

When examining register or memory data, you can view the data in binary, hex, or decimal format. Just use the "Register" pull down menu to select.

Kernel Text and Kernel Data may not be necessary to be viewed all the times, you can unselect them by unselecting "Kernel Text" in the "Text Segment" pull down menu and unselecting "Kernel Data" in the "Data Segment" pull down menu.

You can set breakpoints in your code simply by right clicking on an instruction in the Text tab.

To view memory data, simply click on the Data tab.

By right clicking on a register file value or memory address value, you can change its contents dynamically.

Example Program

Below is an example program to find the sum of an array. Copy this into a text editor and save it as a .s file and open it in QtSpim by loading the file. You can directly run it or do single stepping and observe the change in the Register file. At the end of the Program you should be able to see the result stored in S1 as "1e" (2+4+6+8+10 = 30 = 0x1e) and the console will print this result. The code is well commented which should help you start straight away.,

# first SPIM program	
# ECE 484/584 #	
.data	# Put Global Data here
N: .word 5	# loop count
X: .word 2,4,6,8,10	# array of numbers to be added'
SUM: .word 0	# location of the final sum

str:

```
.asciiz "The sum of the array is = "
```

.text	# Put program here
.globl main	# globally define 'main'
main: lw \$s0, N	# load loop counter into \$s0
la \$t0, X	# load the address of X into \$t0
and \$s1, \$s1, \$zero	# clear \$s1 aka temp sum
loop: lw \$t1, 0(\$t0)	# load the next value of x
add \$s1, \$s1, \$t1	# add it to the running sum
addi \$t0, \$t0, 4	# increment to the next address
addi \$s0, \$s0, -1	# decrement the loop counter
bne \$0, \$s0, loop	# loop back until complete
sw \$s1, SUM	# store the final total
li \$v0, 10	# syscall to exit cleanly from main only, refer to Figure # A.9.1 in the Assemblers, Linkers, and the SPIM #Simulator document (PDF).
syscall	# this ends execution
.end	

Steps:

1. Load the program

We were were were were were were were we	rine simulator registers resisegment Data segment Window Help				
Integr					
The real Sequence (DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotADDOD)(DotA	IPtegs Int Regs [16]	Data	Text		
Lander C. Martinkaner Kanchakaner Manner (J. Jamp, 2003) Wirksmann, J. and Kanchakaner J. Statistick (J. Jamp, 2003) Al Kapita kenner J. Jamp, J. Kanchakaner J. Jamp, J. Kapita (J. Jamp, 2003) Wirksmann, J. Jamp, J.	Attempt Attempt are 0 are 0 statu - are 0 this icon - base -	* #1 (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000) (0040000)	Bfa40000 im 84, 0(829) The5004 Addim 85, 839, 4 0005100 ell 82, 84, 3 0005100 ell 82, 84, 3 0000000 pull 82, 84, 1 0000000 pull 82, 84, 10 0000000 err 87, 80, 10 0000000 err 87, 80, 10	Umer Text Segment [0040000][0044000] / 1857 ir fat C(fag) # argo / 1867 radiu 61 Sp # farge / 1867 radiu 24 Sp # farge / 1867 radiu 24 Sea Pro / 1867 radiu 24 Sea P	
	Ladde C. Charlen Indersteigenstal konffreidig, sense pdf824 Sense and Sense S				
SPR Verson 1. 4 of Speeter 4. 2011 Copyred 1939 (2012). A series f. Leus. Al Rights Reserved.	SPIP Wrison 9, 1.4 of September 4, 2011 Copyright 1990-2020, James R. Larus. All Rights Reserved.				

2. Execute

<pre>Mile The Part of the Part</pre>	Ar QrSpim		Particular Annual Annual A.A.	00
Preprint Image: Ima	rie simulator registers les segment trats segment mindow ries	*		
In the line (10) Use the the line (10) In the the line (10) In the the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) In the line (10) I				
The Tax Segue (100000)(00000) The Tax Segue (10000)(00000) The Tax Segue (10000)(0000) The Tax Segue (10000)(0000)(0000) The Tax Segue (10000)(0000) The Tax Segue (10000)(0000)(0000) The Tax Segue (10000)(0000)(0000) The Tax Segue (10000)(0000)(0000) The Tax Segue (10000)(0000)(0000) The Tax Segue (10000)(0000)(0000)(0000) The Tax Segue (10000)(0000)(0000)(0000)(0000)(0000)(0000) The Tax Segue (10000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000)(0000).	IP Regs Int Kegs [16]	Data Text		
<pre>pre</pre>	pre-renge [Le]		These Part Research (PAJPARAN) (PAJJARAN)	
Image:	EPC = 0	[00400000] Sfa40000 1w \$4, 0(\$29)	7 1837 1w SaO C(Sap) # argo	
<pre>kink to interview int</pre>	Cause = 0	[00400004] 27a50004 addim \$5, \$29, 4	: 184r addiu Sal Sap 4 # argv	
<pre>Automatic account of the second of the</pre>	BadVAddr = 0	[00400008] 24260004 addim \$6, \$5, 4	/ 188: addiu Sa2 Sa1 4 # envp	
<pre>mt _ c · c · c · c · c · c · c · c · c · c</pre>	Status - JUUVALLU	[0040000c] 00041080 #11 \$2, \$4, 2	/ 1847 sll SvO SaO J	
<pre>i - o - o - o - o - o - o - o - o - o -</pre>	HI - 0	[00400016] 00223021 Addm 96, 95, 92	/ LOVE doubled and and and	
<pre>mp to 1 = 0 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 / 1/2 /</pre>	LO - 0	[00400018] 00000000 mop	/ 1881 nop	
<pre>http://wickersensessessessessessessessessessessessess</pre>	P0 (r0) = 0	[0040001c] 3402000a ors \$2, \$0, 10	7 1917 11 Sv0 10	
<pre>bit bit bit bit bit bit bit bit bit bit</pre>	RI $[at] = 0$	[00400020] 0000000c syscall	/ 192: syscall # syscall 10 (exit)	
<pre>http://www.statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/statics.com/st</pre>	R2 [v0] = 0	[00400024] 3c011001 lus \$1, 4097	s 241 Iw 580, W # load loop counter into 580	
<pre>http://www.set.org/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/interface/in</pre>	R3 (V1) = 0 Hit the Run	[00400020] SC300000 10 \$16, 0(\$1)	· Set to Bed. W.B. Load the address of V jato Std	
<pre>ki ki k</pre>	R4 [a0] = 2	[004000301 34280004 or1 58, 51, 4 [X]		
<pre>xi xi zi o c m refrorm single stepping</pre>	R6 (a2) = 7ffffee	[00400034] 02208824 and \$17, \$17, \$0	/ 281 and Sal, Sal, Spero # clear Sal aka temp sum	
<pre>mt to 1 = 0 mt to 1 = 0 m</pre>	R7 [a3] = 0	[00400038] 8d090000 lw \$9, 0(\$8)	: 30: 1v Sti, 6(StO) # load the next value of x	
<pre>pi ti i o or perform singe pi ti i o or perform singe stepping</pre>	R8 [t0] = 0	[0040003c] 02298820 add \$17, \$17, \$9	/ 32: add 3s1, 8s1, 8t1 # add it to the running sum	
his 1 (si = 0 stepping 1 (si = 0 1 (si = 0) 1 (si	is its is or perform single	[00400040] 21080004 add1 58, 58, 4	7 347 addi 550, 540, 4 # increment to the next address	
N12 141 = 0 Jump / 100 110 is 15, 607 2 for find, 100 111 is 15, 607 2 for find, 100 111 is 10, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 607 2 for find, 100 111 is 100, find, 600 111 is 100, fi	R11 (13) = 0 stenning	(00400048) 1410fffc bme \$0, \$16, -16 [104	en-0x00400481	
<pre>Hit 1 10 = 0 Hit 1 10 = 0</pre>	B12 [14] = 0	[0040004c] 3c011001 lui \$1, 4097	: 40: sv 5s1, SUM # store the final total	
Int 1 (10) = 0 (10) (10) = 0 (10) (10) (10) (10) (10) (10) (10) (10)	R13 [15] = 0	[00400050] ac310018 aw \$17, 24(\$1)		
<pre>http://www.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.interference.in</pre>	H14 [16] = 0	[00400054] 36020004 ori \$2, \$0, 4	/ 43: li \$v0, 4 # system call code for print_str	
<pre>Nr 1 k1 - 0 k1 + 0</pre>	R16 [s0] = 0	[00400058] 3c011001 1mi \$1, 4097 [str]	7 45: la Sao, str # address of string to print	
110 110 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 1	B17 [s1] = 0	[004000601 0000000c avecal1	/ 47: syscall	
11/2 11/2 1 -0 12/2 12/3 -0 -0 12/2 12/3 -0 -0 12/2 12/3 -0 -0 12/2 12/3 -0 -0 -0 12/2 12/3 -0 -0 -0 -0 12/2 12/3 -0 -0 -0 -0 -0 12/2 12/3 -0 -0 -0 -0 -0 -0 12/2 12/3 -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 -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	R10 [s2] = 0	[00400064] 34020001 ors \$2, \$0, 1	: 80: 11 \$v0, 1 # system call code for print str	
Intel = 0 Rol 161 = 0 R	R19 (s3) = 0	[00400068] 3c011001 lut \$1, 4097	; 52: Iv SaO, SUM # Print the SUM	
120 100 00000000 pressli 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	$R_{21} = 0$	[0040006c] 8c240018 1w \$4, 24(\$1)		
ID3 I	R22 [s6] = 0	- [00400070] 00000000 syscall	/ E4/ syscall	
Idd 18 - 0	R23 [s7] = 0	[00400074] 34020008 OF1 82, 80, 10	/ Set is you, by a system to exit classing from main only	
	$R_{24}[t0] = 0$	footooral occorrect alacart	, ev. spear + one and execution	
Not 2 11 - 0 - Not 2 11 - 0 - State 1 - 0 - State 2 - 0 - <	$R_{20} [t_{9}] = 0$			
<pre>index Classification Classifica</pre>	$R_{27}(k1) = 0$	*		
Silv Version 51. set September 4. 201 Convert 199-20. All September 4. 201 Silv Version 51. set S	Loaded: C. Liters Intudent Accounted and Templet temp. col9214			(*
	SPDM Version 9.1.4 of September 4, 2011			
SPU is de la de construction de la la construction de la construction	Al Rights Reserved.			
An it is more and registrance.	SPDH is distributed under a BGD license.			
henory and registers dendel Lossel Culture Human Henory and Henory Lange Henory 2014 Enory 11 of Registers Analy Algebra Reservation See the Residence for All Conjugation See Conjugation See See Reservation See See See See See See See See See Se	see are ne rescher to a far copyright roote.			
	Memory and registers deared			
SPAR second A fold Research A	Loaded: C:/Usera/student/AppData/Local/Temp/ot_temp.MH6924			
	SPDH Version 9, 1, 4 of September 4, 2011 Conversiti 1990-2013, Januar R. Larus,			
	All Rights Reserved.			
	SPDK is distributed under a EGD license. See the file README for a full copyright notice.			
	Memory and registers cleaned			
	Loaded: C: (Users/student/AppClata/Local/Temp/of_temp.Mc6924 SP3M Version 9, 1, 4 of September 4, 2011			
Alapa Rennes.	Copyright 1990-2010, James R. Larus.			
	All Rights Reserved.			
		I want I II will will work with and		
	🐴 (C) D) 🖺 👩 🚨 📐			- P D (0 104 PM

3. Observe the change in Register contents

<pre>style to set to se</pre>	tReps [24] IC = 400044 3C = 0	8 × Text			
<pre> transfer t</pre>	C = 400044 PC = 0				8
The second seco	1PC = 0			User Text Segment [00400000][00440000]	
<pre>seture = interpretender interpr</pre>		[00400000]	fa40000 1w \$4, 0(\$29)	/ 183: 1# \$e0 0(\$sp) # argc	
<pre>km _ soortsol</pre>	adde = 0 adVAddr = 0	[00400004]	7a50004 add1u \$5, \$29, 4	7 184: addiu Sal Sap 4 # argv	
A A A A A A A A A A A A A A A A A	tatus = 3000ff10	[0040000c]	00041080 #11 \$2, \$4, 2	7 1867 MINI SWO BaO 2	
<pre>c c c c c c c c c c c c c c c c c c c</pre>	er notes	[00400010]	0c23021 addu \$6, \$6, \$2	: 107: addu Sal Sal Sv0	
<pre>http://www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/www.itemachan.com/</pre>	.0 = 0	[00400014]	e100009 jal 0x00400024 [main]	2 1882 jal main	
<pre>u of of</pre>		[00400016]	4020000 nop	7 1697 Rop 2 1912 14 990 10	
<pre>pipe pipe pipe pipe pipe pipe pipe pip</pre>	0 (r0) = 0	[00400020]	0000000c syscall	/ 192: syscall # syscall 10 (exit)	
<pre>a for i = 1 for i = 1</pre>	12 [v0] = 0	1004000341	collool lui \$1, 4097	: 24: 1v Ss0. N # load loop counter into Ss0	
<pre>4 dig i dig i</pre>	a [v1] = 0	(00400028)	C300000 1w \$16, 0(\$1)	- 26. In \$th W # load the address of W into \$th	
<pre>bit bit bit bit bit bit bit bit bit bit</pre>	A [a0] = 2	[00400030]	14280004 ori \$8, \$1, 4 [X]	1 Mai 14 Apr, W & Toan The addless of W Turn and	
<pre>1 (1 1 - 0 - 1 - 0 - 1 - 0 - 1 - 0 - 0 - 0</pre>	6 [a2] = 7ffff628	[00400034]	2205824 and \$17, \$17, \$0	: 28: and \$s1, \$s1, \$zero # clear \$s1 aka temp sum	
<pre>i di i d</pre>	7 (a3) = 0 Observe the chan	ge in [00400038]	1d090000 1w \$9, 0(\$8)	; 30: 1# \$t1, 0(\$t0) # load the next value of x	
<pre>up 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>	(t0) = 10010008	1004000301	1080004 add \$5, \$8, 4	7 Jul and Jul, Jul, Jul # and it to the running sum 2 Jul addi Sto. Sto. 4 # increment to the next address	
<pre>11 [1] 1 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0</pre>	ao (t2) = 0 contents of \$51	1004000441	210ffff addi \$16, \$16, -1	7 36; addi \$s0, \$s0, -1 # decrement the loop counter	
<pre>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</pre>	11 [13] = 0	[00400048]	410fffc hoe \$0, \$16, -16 [loop	-0x00400048)	
<pre>Li li li</pre>	12 [64] = 0	[0040004c]	CO11001 101 \$1, 4097	7 407 sw Ssl, SUM # store the final total	
<pre>11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</pre>	114 [t6] = 0	[00400054]	4020004 ort \$2, \$0, 4	: 43: 11 3v0, 4 # system call code for print str	
Al 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1	us (tr) = 0	[00400058]	e011001 lui \$1, 4097 [str]	/ 48: la SaO, str # address of string to print	
Lie food of the priority area for the state of the state	16 [80] = 5	{0040005c}	424001c ori \$4, \$1, 28 [str]		
<pre>13 [13] = 0 [14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0 \\[14] = 0</pre>	18 [s2] = 0	[00400060]	4020001 or1 \$2.50 1	7 4/7 Systemi - Sdt 1/ Sud. 1 # sustan call code for opint ste	
U = [43] = 0 U = [43] = 0 U = [43] = 0 U	u9 [83] = 0	[00400068]	e011001 lui \$1, 4097	; 52: 19 \$40, SUM # Print the SUN	
22 14 2 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	20 [s4] = 0 (21 (s5) = 0	[0040006c]	c240018 1w 84, 24(\$1)		
<pre>201 [02] 0 0 0 0 0 0 0 0 0 0</pre>	22 [86] = 0	[00400070]	coccoc syscall	7 547 syscall	
de la	23 [87] = 0	1004000781	000000c syscall	: do: syscell # this ends execution	
De g lag e g e g Jag e jag e j Jag e j	24 [te] = 0 25 [te] = 0	S2 - 25	100		
12 trul 1 = 0 v v v v v v v v v v v v v v v v v v	26 [k0] = 0				
and i C (an	27 [k1] = 0				
school of series (Lines) school of series (taded: C:/Users/student/AppCats/Local/Temp/qt_temp-pq6924				
Hand Band Series Hand Band Series Hand Ser	opyright 1990-2010, James R. Larus.				
Hardy and righter dans. Hardy and righter dans. Hardy and righter dans. Hardy and righter dans. Hardy hards to any dans the Laws. Hardy hards to any dans the dans. Hardy hards to any dans the dans. Hardy hard to be the dans dans the dans. Hardy hards to any dans the dans. Hardy hard to be the dans dans the dans. Hardy hard to be the dans dans dans dans dans dans dans dans	J Rights Reserved. #2M is distributed under a 85D license.				
Menor yor inglem deed PP Negro A. Loo Devolution, Loop, And 20 PP Negro A. Loop, And 20 PP Negro	ee the file README for a full copyright notice.				
Mary & forgers dans Samper for some of the source of the					
PPP Interior 1, set of Deprese PPP Interior 1, set of Deprese	amory and registers deared paded: C-Users/student/AppCata/Local/Temp/ot_temp.MH6924				
and grad and	PIM Version 9.1.4 of September 4, 2011				
094 & & Define dend 6 & REDECT & Manypell notes: 4 Any Part Res 1 6 & REDECT & Manypell notes: 4 Any Part Res 1 6 &	a Rights Reserved.				
Menary and Hoppins Gared Society Classification (Classification (Classification Classification	20H is distributed under a BSD license. we the file README for a full copyright notice.				
leen y af rights dared and C. Uwer Marken Markel Manage Man Markel Storg At North Storg At					
ander Concernen La La Anders Angele Concernen La La Concernen La Con	lemony and registers cleared				
ადიტმ 10000, Jane R. Lana Kingd Napur ლიდ Sapu ლიდ Sapu ლიდ	saded: ClUbers/student/Appoata/Loca/temp/qt_temp.Mct/424 PIM Version 9.1.4 of September 4, 2011				
Sen Sep	opyright 1990-2010, James R. Larus.				
S C D 🖹 🖗 🗱 📐 S 4 6/ 33 W 🖹	ingle Step				
		T HAN TO T			310 PM
			🖅 🙄 🛅 🔛		- P 10 10 1/6/2012

File Smulator Registers Set Genneet Data Genneet Window Hele				
FP Kegs Inc Kegs [10]	Duca re	at		
FPC = 0	1004000001 #5+40000	14 64 046201	. 169: Is fail 0/femi # amm	
Cause = 0	[00400004] 27450004	addiu \$5, \$29, 4	z 184: addin Sal Sap 4 # arov	
BadVAddr = 0	[00400008] 24460004	addiu \$6, \$5, 4	: 185: addiu Sal Sal 4 # envp	
Status = 3000ffl0	[0040000c] 00041080	#11 52, \$4, 2	7 186: sll Sv0 Sa0 2	
and Dates	[00400010] 00c73023	addu \$6, \$6, \$2	: 187: addu Sa3 Sa2 Sv0	
10 = 0	[00400014] 0c100003	jal 0x00400024 [main]	z 186: jal mein	
	[00400018] 00000000	nop	7 1891 nop	
R0 (r0) = 0	[0040001c] 3402000s	ori \$2, \$0, 10	/ 191/ 11 \$V0 10	
R1 [at] = 10010000	[00400020] 00000000	ayscall	/ 192: Ayscall # Syscall 10 (exit)	
R2 [v0] = a	(004000281 8c300000	1w \$16. D(\$1)	. set as tas, a story we connect the say	
R3 [V1] = 0	(0040002c1 3c011001	1ui \$1, 4097 [X]	7 267 1s StO. X # Load the address of X into StO	
$\frac{1}{10} \frac{1}{10} = \frac{1}{10} \frac{1}{10}$	[00400030] 34280004	ori \$8, \$1, 4 [X]		
R6 [a2] = 7ffff628	[00400034] 02205824	and \$17, \$17, \$0	: 28: and \$s1, \$s1, \$zero # clear \$s1 aka temp sum	
R^{7} (s3) = 0 2+4+6+8+10 = 30 = 0×1e	[00400038] 84090000	1w \$9, 0(\$0)	; 30: 1v \$t1, 0(\$t0) # load the next value of x	
R8 [t0] = 10010018	[0040003c] 02298820	add \$17, \$17, \$9	/ 32/ add Ssl, Ssl, Stl # add it to the running sum	
R9 [t1] = a	1004000401 31080004	add1 \$8, \$9, 4	/ 34: addi \$50, \$50, 4 # increment to the next address	
R10 (12) = 0	[00400044] 2210EEE	addi 316, 516, -1	7 367 addi 250, 250, -1 # decrement the loop counter	
812 (141 = 0	(0040004c] 10011001	Int \$1, 4097	/ doi av Sel. SIM # store the final total	
R13 (15) = 0	[00400050] ac310018	aw \$17, 24(\$1)		
R14 [t6] = 0	[00400054] 34020004	ort \$2, \$0, 4	3 43: 11 5v0, 4 # system call code for print str	
R15 [t7] = 0	[00400058] 3c011001	lui \$1, 4097 [str]	/ 46: la SaO, str # address of string to print	
R16 [s0] = 0	[0040005c] 3424001c	ori \$4, \$1, 28 [str]		
B10 [02] = 0	[D04000e0] 00000000	syscall	r 47r systell	
B19 (e3) = 0	[00400044] 34020001	or1 \$2, \$0, 1	r SOr II Sv0, 1 # system cell code for print_str	
R20 [s4] = 0	[00000068] 30011001 (00000661 86240018	101 51, 4097	/ 52: 19 FAC, SUN # Print the SUN	
R21 [s5] = 0	1004000701 00000000	avacal1	7 54: syscall	
R22 [s6] = 0	1004000741 34020004	or1 \$2, \$0, 10	7 58; 11 5v0, 10 # syscall to exit clearly from main only	
R23 [#7] = 0	[00400078] 00000000	syscall	/ 60; syscall # this ends execution	
$R_{24} [t_{0}] = 0$ $R_{25} [t_{9}] = 0$	S			
826 (k0) = 0				
R27 [k1] = 0	*			
Looped - C 1 June 14 der 20 Constant June (Temp) et _emp.get/924 2020 Henrich 11 - C 4 Senableche + C 14 Capying Harris 1950-2010, James R. Larus, All Rights Ream- Rights Ream- Rights Ream- Set Park RE2020 Henric Set Park RE2020 Henric Se				
Nencoy and register deared Londer C. Johan Hunder NepOlat Local Respirit Jerry MH6924 SPV Haron S. 1.4 disgosterior H. 2011 Copyright 1980-230, James B., Luna, Alfordin Hearner, DPN et distributed undre a KID Norse. Set the the RESEACH for a Kill copyright notice.				1
Nemory and registers cleared Loaded: C.(Lierskindert/ApOLita), Loal/Templot, temp.Mc6924 93/W version 3, L. 46 September 4, 2011 Covyright 1990-0010, James R. Lanus. All Rights Reserved.				

- P- 12 40 311 PM

🔞 🙆 🔉 🚆 🗿 🛎 🔎 🕵 🐛 🛷 🤤 👑 🐯