Computer Architecture

SPIM Simulator

Made by Park, Byung-choon

LIST

- SPIM Definition
- Multiple versions
- Memory Layout
- PCSPIM
- Example
- Homework

SPIM

- Definition
 - SPIM is a software simulator that runs programs written for MIPS R2000/R3000 processors
 - SPIM can read and immediately execute assembly language files or MIPS executable files
 - SPIM is a self-contained system
 - Debugger
 - A few operating system-like services

SPIM of Multiple versions

SPIM

- Command-line-driven program
- Requires only an alphanumeric terminal to display

XSPIM

- X-windows environment
- Much easier program to learn

PCSPIM

• Windows version of SPIM

Memory Layout



PCSPIM

Download and Install

Windows

- Register display
- Text segments
- Data and stack segments
- SPIM messages
- Function
 - Load
 - Go
 - Single step
 - Multiple steps
 - Breakpoint

PCSPIM Download

Downloading PCSPIM

• <u>http://www.cs.wisc.edu/~larus/spim.html</u>

Downloading SPIM

Platform	Program	Form	File
Unix or Linux system	spim	Source code	http://www.cs.wisc.edu/~larus/SPIM/spim.tar.Z or
Mac OS X	xspim		http://www.cs.wisc.edu/~larus/SPIM/spim.tar.gz
Timuz	spim	Binary RPM	http://www.cs.wiec.edu/chi/downloads/
	xspim	for Fedora	Intp.//www.cs.wisc.cdu/cb/dowinoads/
Microsoft Windows		_	
(Windows NT, 2000, XP)		Executable	http://www.cs.wisc.edu/~larus/SPIM/pcspim.zip
	spim		
(spim 7.0 and later versions no	PCSpim		
longer run on Windows 95/98.		Source code	http://www.cs.wisc.edu/~larus/SPIM/pcspim_src.zip
Use version 6.5 or earlier.)			

Executable File Install

😂 pospim		
파일(E) 편집(E) 보기(⊻) 즐겨최	\$기(<u>A</u>) 도구(I) 도움말(<u>H</u>)	
🕝 뒤로 🔹 🕥 🍷 🏂 🔎 검	색 💫 폴더 💷 -	
주소(D) 🛅 C:₩Documents and Sett	ings₩jspark₩UŀEr호ŀEj₩pcspim	💟 🄁 이동
파일 및 폴더 작업 🔹	setup.exe Setup	
▲ 폴더를 웹에 게시 ☆ 폴더 공유	Setup,msi Windows Installer 패키지 1,257KB	
기타 위치 📎		
ករងត់ 📚		
pcspim 파일 폴더 수정한 날짜: 2009년 4월 15 일 오늘, 오전 9:11		
0.711-511		
2 개제	1,6/MB 😏 내 컴퓨터	

Step 1



n PCSpim					
파일(F) 편집(<u>E</u>) 보기(<u>V</u>) 즐	·겨찾기(<u>A</u>) 도구(<u>T</u>)	도움말(<u>H</u>)			11
실퇴로 • ➡ • 🔂 🙆 검색	Ra≝ri Øål P3 D3	Xn .			
		8 / - 5 1923 -			
					♥ 비용
	이름 🔺	크기	종류	수정한 날짜	▲
	h) ConsoleWnd, h	3KB	C 헤더 파일	2002-01-05 오후	
DOO I	📑 MainFrm.cpp	6KB	C++ 원본 파일	2002-01-05 오후	
PCSpim	h MainFrm, h	3KB	C 헤더 파일	2002-01-05 오후	
	📑 MakeHelp bat	2KB	MS-DOS 일괄 파	일 2002-01-05오후	
	📾 msscoprj, soc	IKB	SUC 파일	2002-01-05 오후	
프로젝트 작립 공간	MultiStepvig.cpp	ZKB	U++ 원본 파일 이해다 피아	2002-01-05 오후	
수정한 날짜: 2002-01-05 오후 2:0)5 h MultiStepvig, n	ZKB	니에너 파일 ADO 파이	2002-01-05 오후	
	PCSpim, aps	53NB CKB	APS 파일 이 씨 파이	2003-01-04 오후	
크기: 6498[0] 드	PCSpim,ciw		CLW 파일 CNT 파이	2003-01-04 오우	
특성:(일반)	PCSpim,cnt		UNT 파일 C., 이번 파이	2002-01-05 포우	
	CSpim,cpp		U++ 원본 파일 고글제도 파이	2002-01-05 포우 2002-01-04 0전	
	間hcshill,ush		프로젝트 파월 고쿄제도 자연 고기	2003-01-04 또신	
	B DCSpin b		그도쪽도 역합 등;	신 2002-01-05 포우 2002-01-05 오늘	
	<u>п</u> нозріні, н Авосерім ці р	ם ארג משקר	C에너 파골 드오마 페이	2002-01-03 포우 2000-12-24 오늘	
	S poopim mdp		포함을 따을 MDD 피어	2000-12-24 포우 2002-01-05 오늘	
	📄 pospim, nup	40KD 73KB		2002-01-03 또루 2002-01-03 또루	
	a pospim, not	55K B	이미지 파이	2003-03-31 또한	
	a pospim pla	2KB	HTML Document	2003-01-04 1 +	
	PCSpim.rc	19KB	리소스 텐플리	2003-01-04 오늘	
		3KB	C++ 위보 파잌	2002-01-05 오늘	
	b) PCSpimDoc b	2KB	C 체터 파악	2002-01-05 오늘	
	PCSnimView.con	32KB	C++ 원본 파잌	2002-01-03 오후	
	b) PCSnimView h	5KB	이 한 한 한 한 한 한 한 한 한 한 한 한 한 한 한 한 한 한 한	2003-01-04 오흐	
	🗐 readme.txt	2KB	텍스트 문서	2002-01-05 오후	
	🖬 Regman.cop	7KB	C++ 원본 파일	2002-01-05 오후	
	h Regman.h	6KB	C 헤더 파일	2002-01-05 오후	
	h) resource.h	4KB	C 헤더 파일	2003-01-04 오전	
	🖬 resource, hm	1KB	HM 파일	2002-01-05 오후	
	🖬 RunDig.cpp	4KB	C++ 원본 파일	2002-01-05 오후	
	h RunDlg,h	2KB	C 헤더 파일	2002-01-05 오후	
	💼 SettingsDlg,cpp	7KB	C++ 원본 파일	2003-01-04 오전	_1
		סער	CDD πιοι		<u> </u>
송류: 프로젝트 작업 공간 크기: 649	바이트			6498만이트 📙 내 컴퓨	E1 //.

🦇 pospim – Microsoft Visual C++ – [C:\.,.\WMFC\SRC\APPMODUL.CPP] 🗈 File Edit View Insert Project Build Tools Window Help 🎦 🚅 🖬 🕼 🐰 📭 💼 🕰 - 오 - 오 - 🖪 🗖 😽 🐂 - 🐜 Step 2 💌 [All global members 🔍 [No members - Create New Class...] 🔍 📉 🔹 🏙 👗 🚦 🗒 (Globals) #undef THIS FILE A X static char THIS_FILE[] = __FILE__; 🚯 Workspace 'pcspim': 1 proje #endif 🖻 📴 PCSpim files 🗄 🔄 Source Files BreakpointDlg.cpp // export WinMain to force linkage to this module 🖹 ConsoleWnd.cpp extern int AFXAPI AfxWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, 📩 data.c LPTSTR 1pCmdLine, int nCmdShow); 📩 display-utils.c 📩 inst.c extern "C" int WINAPI 📩 lex.yy.c tWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, 📩 MainFrm.cpp LPTSTR 1pCmdLine, int nCmdShow) 📩 mem.c 📩 mips-syscall.c // call shared/exported WinMain MultiStepDlg.cpp return AfxWinMain(hInstance, hPrevInstance, lpCmdLine, nCmdShow); PCSpim.cpp 3 🖹 PCSpim.hpj PCSpim.rc // initialize app state such that it points to this module's core state PCSpimDoc.cpp PCSpimView.cpp BOOL AFXAPI AfxInitialize(BOOL bDLL, DWORD dwVersion) 🖹 Regman.cpp { 📩 run.c AFX MODULE STATE* pModuleState = AfxGetModuleState(); RunDlg.cpp pModuleState->m bDLL = (BYTE)bDLL; ASSERT(dwVersion <= _MFC_VER); SettingsDlg.cpp UNUSED(dwVersion); // not used in release build SetValueDlg.cpp **#ifdef** AFXDLL 📩 spim-utils.c pModuleState->m_dwVersion = dwVersion; SpimSupport.cpp #endif StdAfx.cop **#ifdef** MBCS 📩 sym-tbl.c // set correct multi-bute code-page for Win32 apps 📩 util.cpp if (!bDLL) 📩 y.tab.c _setmbcp(_MB_CP_ANSI); #endif // MBCS 🗄 💼 Header Files return TRUE; 🗄 💼 Resource Files 🗄 parser.y 🖹 scanner.l // force initialization early 🗄 📄 External Dependencies #pragma warning(disable: 4074) #pragma init_seg(lib) **#ifndef** AFXDLL void AFX CDECL AfxTermAppState() // terminate local data and critical sections AfvToemLocalData/NULL TRUEL. 📲 Clas... 👹 Res... 🖹 FileV...





& PCSpim	
Eile Simulator Window Help	
PC = 00000000 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000 Status = 3000ff10 HI = 00000000 LO = 00000000 General Registers 00000000 R24 (t8) = 00000000 R24 (t8) = 00000000 R27 (t1) = 00000000 R1 (at) = 00000000 R17 (s1) = 00000000 R25 (t9) = 00000000 R26 (k0) = 00000000 R27 (k1) = 00000000 R2 (v0) = 00000000 R10 (t2) = 00000000 R18 (s2) = 00000000 R26 (k0) = 00000000 R27 (k1) = 00000000 R3 (v1) = 00000000 R11 (t3) = 00000000 R27 (k1) = 00000000 R28 (gp) = 10000000 R4 (a0) = 00000000 R12 (t4) = 00000000 R21 (s5) = 00000000 R29 (sp) = 7ffffaa8 R6 (a2) = 7ffffab0 R14 (t6) = 00000000 R22 (s6) = 00000000 R30 (s8) = 00000000 R7 (a3) = 00000000 R15 (t7) = 00000000 R23 (s7) = 00000000 R31 (ra) = 00000000	- Register display
Ox00400000] 0x8fa40000 lw \$4,0(\$29) ; 183: lw \$a0 0(\$sp) # argc [0x00400004] 0x27a50004 addiu \$5, \$29, 4 ; 184: addiu \$a1 \$sp 4 # argv [0x00400006] 0x24a60004 addiu \$6, \$5, 4 ; 185: addiu \$a2 \$a1 4 # envp [0x00400006] 0x00401808 s11 \$2, \$4, 2 ; 186: s11 \$v0 \$a0 2 [0x00400010] 0x00c23021 addu \$6, \$6, \$2 ; 187: addu \$a2 \$a2 \$v0 [0x00400014] 0x0c000000 ja1 0x00000000 [main] ; 188: jal main [0x00400016] 0x3402000a ori \$2, \$0, 10 ; 191: li \$v0 10 [0x00400020] 0x0000000c syscall ; 192: syscall # syscall 10 (exit)	- Text segments
	_
DATA [0x10000000][0x10040000] 0x00000000	
[0x7ffffaa8] 0x0000000 0x0000000	- Data segments
KERNEL DATA [0x9000000] 0x78452020 0x74706563 0x206e6f69 0x636f2000 [0x9000010] 0x72727563 0x61206465 0x6920646e 0x726f6e67 [0x9000020] 0x000a6465 0x495b2020 0x7265746e 0x74707572 [0x9000030] 0x2000205d 0x4c545b20 0x20005d42 0x4c545b20 0x20005d42	
SPIM Version 8.0 of January 8, 2010 Copyright 1990-2010, James R. Larus. All Rights Reserved. DOS and Windows ports by David A. Carley. Copyright 1997, Morgan Kaufmann Publishers, Inc. See the file README for a full copyright notice. Loaded: C:\Program Files\PCSpim\exceptions.s	- SPIM messages
For Help, press F1 PC=0x00000000 EPC=0x00000000 Cause=0x000000000	

Register display

It shows the values of all registers in the MIPS CPU and FPU.

4	🔉 PCS	oim								
ļ	<u>F</u> ile <u>S</u> ir	nulator	r <u>W</u> indow <u>H</u> e	lp						
I	2		D 🕘 🥐 🕅	?						
Γ	PC	=	00000000	EPC	=	00000000	Cause	= 00000000	BadVAddr= 00000000	~
	Statu	s =	3000ff10	HI	=	00000000	LO	= 00000000		
F						General	Registers			
F	≀0 (r	0) =	00000000	R8 (t0	0) =	00000000	R16 (sO)	= 00000000	R24 (t8) = 00000000	
F	{1 (a	t) =	00000000	R9 (t:	1) =	00000000	R17 (s1)	= 00000000	R25 (t9) = 00000000	
F	≀2 (v	0) =	00000000	R10 (t2	2) =	00000000	R18 (s2)	= 00000000	R26 (k0) = 00000000	
F	≀3 (v	1) =	00000000	R11 (t3	3) =	00000000	R19 (s3)	= 00000000	R27 (k1) = 00000000	
F	₹4 (a	0) =	00000000	R12 (t4	4) =	00000000	R20 (s4)	= 00000000	R28 (gp) = 10008000	
F	≀5 (`a	1) =	00000000	R13 (ts	5) =	00000000	R21 (s5)	= 00000000	R29 (sp) = 7ffffaa8	
F	₹6 (a	2í =	7ffffab0	R14 (t6	6j =	00000000	R22 (s6)	= 00000000	R30 (sð) = 00000000	
F	₹7 (a	3) =	00000000	R15 (t7	7) =	00000000	R23 (s7)	= 00000000	R31 (ra) = 00000000	~
	<									>

General-purpose registers

Text segments

Displays instructions both from your program and the system code that is loaded automatically when PCSPIM starts running

$ \begin{bmatrix} 0 \times 00400000 \\ 0 \times 00400004 \\ \end{bmatrix} \\ \begin{bmatrix} 0 \times 00400006 \\ 0 \times 0040000c \end{bmatrix} \\ \begin{bmatrix} 0 \times 00400010 \\ 0 \times 00400014 \end{bmatrix} \\ \begin{bmatrix} 0 \times 00400018 \\ 0 \times 0040001c \end{bmatrix} $	0x8fa40000 0x27a50004 0x24a60004 0x00041080 0x00c23021 0x0c000000 0x00000000 0x3402000a	<pre>lw \$4, 0(\$29) addiu \$5, \$29, 4 addiu \$6, \$5, 4 sll \$2, \$4, 2 addu \$6, \$6, \$2 jal 0x00000000 [main] nop ori \$2, \$0, 10</pre>	; 183: lw ŞaO O(Şsp) ; 184: addiu Şa1 Şsp 4 ; 185: addiu Şa2 Şa1 4 ; 186: sll ŞvO ŞaO 2 ; 187: addu Şa2 Şa2 ŞvO ; 188: jal main ; 189: nop ; 191: li ŞvO 10	# argc # argv # envp
[0x00400020]	0x0000000c	syscall	; 192: syscall	# syscall 10 (exit)



Computer Architecture & Network Lab

Data segments (Data and stack segments)

Displays the data loaded into your program`s memory and the data on the program`s stack

SPIM messages

This is where error messages appear

Simulator Function

Load

- File \rightarrow Open
- Select assembly file
- **Reload**
 - File \rightarrow Reload
 - Reload source file after change it with editor program
- 🛛 Go
 - Simulator \rightarrow Go
 - Results are displayed in console
- Single step
 - Simulator \rightarrow Single Step
 - Run an instruction at a time
- Multiple step
 - Simulator \rightarrow Multiple Step
 - Run given number of instruction at a time
- Breakpoint
 - Simulator \rightarrow Breakpoint
 - Stop program immediately before it executes a particular instruction

Function - Load

Elle Simulator Window Help Open Save Log File Ctrl+O Save Log File Ctrl+S Exit Alt+F4 FX = 00000000 General Registers R0 (r0) = 00000000 R1 = 00000000 R2 (r0) = 00000000 R3 (r1) = 00000000 R4 (a0) = 00000000 R4 = 00000000 R2 (x400000 lw \$4, 0 (\$29) (bx00400000] 0x86a40000 0x27a50004 addiu \$5, \$29, 4 (bx00400006] 0x24a60004 x27a50004 addiu \$6, \$5, 4 (bx00400006] 0x20620000 x1 42: addiu \$a1, \$sp, 4 # argc (bx00400006] 0x00020000 (bx00400006] 0x00000000 (bx00400006] 0x00000000 (bx00400016] 0x000
Open Ctdl+O Save Log File Ctdl+S EPC = 00000000 Cause = 00000000 BadVAddr= 00000000 General Registers General Registers 00000000 R1 (at) = 00000000 R8 (tU = 00000000 R16 (s0) = 00000000 R1 (at) = 00000000 R8 (tU = 00000000 R16 (s0) = 00000000 R25 (t9) = 00000000 R2 (v0) = 00000000 R10 (t2) = 00000000 R18 (s2) = 00000000 R26 (k0) = 00000000 R3 (v1) = 00000000 R11 (t3) = 00000000 R19 (s3) = 00000000 R27 (k1) = 00000000 R4 (a0) = 00000000 R12 (t4) = 00000000 R28 (gp) = 10008000 # argc [0x00400000] 0x8fa40000 lw \$4, 0(\$29) ; 140: lw \$a0, 0(\$sp) # argc [0x00400008] 0x24a60004 addiu \$5, \$29, 4 ; 142: addiu \$a1, \$sp, 4 # envp [0x00400008] 0x24a60004 addiu \$6, \$6, \$2 [0x00400010] ja1 0x00000000 [main] [0x00400014] 0x00000000 ja1 0x00000000 [main] mop [main] system1-1.asm [0x0040001c] 0x3402000a ori \$2, \$0, 10 system1-2.asm system1-2.asm
Save Log File Ctrl+S EPC = 00000000 Cause = 00000000 BadVAddr= 00000000 Exit Alt+F4 H1 = 00000000 L0 = 00000000 General Registers 00000000 R24 (t8) = 00000000 R24 (t8) = 00000000 R1 at = 00000000 R9 (t1) 00000000 R15 (s0) = 00000000 R25 (t9) = 00000000 R2 (v0) = 00000000 R10 (t2) = 00000000 R18 (s2) = 00000000 R27 (k1) = 00000000 R27 (k1) = 00000000 R4 (a0) = 00000000 R12 (t4) = 00000000 R20 (s4) = 00000000 R28 (gp) = 10008000 # argv [0x00400001] 0x2fa40000 lw \$4, 0(\$29) ; 140: lw \$a0, 0(\$sp) # argv [0x00400002] 0x0004180 s11 \$2, \$4, 2 ; 142: addiu \$a1, \$sp, 4 # argv [0x00400010] 0x02c32021 addu \$6, \$5, \$4 [* 142: addiu \$a2, \$a1, 4 # envp [0x00400014] 0x00000000 ja1 0x000000000 [main] mop [0x0040001c] 0x3402000a ori \$2, \$0, 10 [* system1-1,asm
Exit Alt+F4 FX = 00000000 LO = 00000000 General Registers R0 (t0 = 00000000 R1 (t1 = 00000000 R1 (t0 = 00000000 R1 (t1 = 00000000 R1 (t1 = 00000000 R1 (t1 = 00000000 R25 (t9) = 00000000 R1 (at) = 00000000 R10 (t2) = 00000000 R18 (s2) = 00000000 R25 (t9) = 00000000 R26 (k0) = 00000000 R2 (v0) = 00000000 R11 (t3) = 00000000 R18 (s2) = 00000000 R26 (k0) = 00000000 R27 (k1) = 00000000 R4 (a0) = 00000000 R14 (t3) = 00000000 R20 (s4) = 00000000 R28 (gp) = 10008000 Image: [0x00400001] 0x27a50004 addiu \$5, \$29, 4 : 140: 1w \$a0, 0(\$sp) # argv Image: [0x00400002] 0x00041080 s11 \$2, \$4, 2 : 141: addiu \$a1, \$sp, 4 # argv [0x00400002] 0x00041080 s11 \$2, \$4, 2 : 142: addiu \$a2, \$a1, 4 # envp Image: [0x00400016] 0x00000000 jadu \$a1 \$c, \$6, \$c, \$2 : 142: addiu \$a2, \$a1, 4 # envp Image: [0x00400016] 0x00000000 jadu \$a1 \$c, \$6, \$c, \$2 : 142: addiu \$a2, \$a1, 4 # envp Image: [0x00400016] 0x000000000 jadu \$x00000000 image: [x140] </td
General Registers R0 (r0) = 00000000 R16 (s0) = 00000000 R24 (t8) = 00000000 R1 (at) = 00000000 R10 (t2) = 00000000 R17 (s1) = 00000000 R25 (t9) = 00000000 R2 (v0) = 00000000 R10 (t2) = 00000000 R18 (s2) = 00000000 R26 (k0) = 00000000 R3 (v1) = 00000000 R11 (t3) = 00000000 R19 (s3) = 00000000 R27 (k1) = 00000000 R4 (a0) = 00000000 R12 (t4) = 00000000 R20 (s4) = 00000000 R28 (gp) = 10008000 I Image: State Stat
【 [0x0040000] 0x8fa40000 lw \$4, 0(\$29) ; 140: lw \$a0, 0(\$sp) # argc [0x0040004] 0x27a50004 addiu \$5, \$29, 4 ; 141: addiu \$a1, \$sp, 4 # argv [0x00400006] 0x20a60004 addiu \$6, \$5, 4 ; 142: addiu \$a2, \$a1, 4 # envp [0x00400010] 0x00c23021 addu \$6, \$6, \$2 [0x00400014] 0x0c000000 ja1 0x00000000 [main] [0x00400018] 0x00000000 nop [0x0040001c] 0x3402000a ori \$2, \$0, 10
[0x00400000] 0x27a50004 addiu \$5, \$29, 4 ; 141: addiu \$a1, \$sp, 4 # argv [0x00400002] 0x24a60004 addiu \$6, \$5, 4 ; 141: addiu \$a2, \$a1, 4 # envp [0x00400002] 0x00041080 sll \$2, \$4, 2 [0x00400010] 0x00c23021 addu \$6, \$6, \$2 [0x00400014] 0x0c000000 ja1 0x00000000 [main] [main] </td
[0x00400008] 0x24a60004 addiu \$6, \$5, 4 ; 142: addiu \$a2, \$a1, 4 # envp [0x0040000c] 0x00041080 sll \$2, \$4, 2 ? [0x00400010] 0x00c23021 addu \$6, \$6, \$2 ? ? [0x00400014] 0x0c000000 jal 0x000000000 [main] ? * ? [0x00400012] 0x00000000 nop
[0x00400010] 0x00c23021 addu \$6, \$4, 2 [0x00400014] 0x00c23021 addu \$6, \$6, \$2 [0x00400018] 0x00000000 jal 0x000000000 [main] [0x00400018] 0x00000000 nop [0x0040001c] 0x3402000a ori \$2, \$0, 10 ▲
[0x00400014] 0x0c000000 jal 0x000000000 [main] [0x00400018] 0x00000000 nop [0x0040001c] 0x3402000a ori \$2, \$0, 10 ・ () * (
[0x0040001c] 0x3402000a ori \$2, \$0, 10 (0x0040001c] 0x3402000a ori \$2, \$0, 10 system1-1,asm
▲ system 1-2, asm
i in system1−3, asm
DATA [Dx10000001[0x100400001 0x00000000
KERNEL DATA III \square
표말 형식(፲):
SPIM Version 6.5 of January 4, 2003
All Rights Reserved.
DOS and Windows ports by David A. Carley (dac@cs.wisc.edu). Copyright 1997 by Morgan Kaufmann Publishers, Inc.
See the file README for a full copyright notice.
Loaded: C:\Program Files\PCSpim\trap.handler
PC-0v0000000 EPC-0v00000000 Cause-0v000000000

Function - Go

À PCSpim		
<u>File</u> <u>Simulator</u> <u>Window</u> <u>Help</u>	r	
Beinitialize		
PC Reload	= 00000000 Cause = 00000000 Bad	VAddr= 00000000
<u> </u>	General Registers	
RO Break	t0, = 00000000 R16 (s0) = 00000000 R24 (t8 t1) = 00000000 R17 (s1) = 00000000 R25 (t9) = 00000000) = 0000000
R2 Single Step F10	t2) = 0000000 R18 (s2) = 00000000 R26 (k0)) = 0000000
R3 <u>M</u> ultiple Step F11 R4	t3) = 00000000 R19 (s3) = 00000000 R27 (k1 t4) = 00000000 R20 (s4) = 00000000 R28 (gp) = 0000000) = 10008000
Breakpoints Ctrl+B		,
Set ⊻alue [Ox(Display symbol table	lw \$4, 0(\$29) ; 140: lw	\$a0, 0(\$sp)
	addiu \$5, \$29, 4 ; 141: add ; addiu \$6, \$5, 4 ; 142: add	iu Şal, Şsp, 4 # argv 🔜 iu Sa2, Sal, 4 # envp
	sll \$2, \$4, 2 ; 143: sll	sv0, sa0, 2
[0x00400014] 0x0c2302) jal 0x00000000 [main] 🔪 🚬 : 144: add	u Şaz, Şaz, ŞVU main
$\begin{bmatrix} 0 \times 00400018 \end{bmatrix}$ $0 \times 000000000000000000000000000000000$	nop Run Parameters	
	Starting	
	0x00400000	
		Caneer
STACK [0x7fffeffc]	0x0000000	
REDUEL DATA	🔽 Check for u	undefined symbol:
KERNEL DATA		
SPIM Version 6.5 of Januar	∼v 4. 2003	4
Copyright 1990-2003 by Jar	nés R. Larus (larus@cs.wisc.edu).	
DOS and Windows ports by I	David A. Carley (dac@cs.wisc.edu).	
Copyright 1997 by Morgan I	aufmann Publishers, Inc.	
Loaded: C:\Program Files\	full copyright notice. PCSpim∖trap.handler	
		v.
٢		
Run the program	PC=0x00000000 EPC=0x00000000 Cause=	0×00000000

Function – Single step

PCSpim	
le <u>S</u> imulator <u>W</u> indow <u>H</u> elp	
같[Clear Registers 	asm 000000
<u>G</u> o Continue	F5 000
Single Step Multiple Step	F10 000 F11 000
Brea <u>k</u> points	Ctrl+B
xt ∑alue xt Display symbol table	p) # argc
Seuritys andorradian (1,2,2,3,1) and (1,2,3,1) and (1,2,3,1) Ix00400010] 0x00c23021 addu \$6, \$6, \$2 ; 144: add Ix00400014] 0x0c100009 jal 0x00400024 [main] ; 145: jal Ix00400018] 0x00000000 nop ; 146: nop Ix0040001c] 0x3402000a ori \$2, \$0, 10 ; 148: li	tu \$a2, \$a2, \$v0 1 main 9 \$v0 10
DATA x1000000][0x10010000] 0x00000000 x10010000] 0x0000000 0x0000065 0x0000000 0x x10010010] 0x3020666f 0x3030317e 0x6d754e20 0x x10010020] 0x000003e 0x0000000 0x0000000 0x x10010030][0x10040000] 0x00000000	x204d5553 x3d726562 x00000000
S and Windows ports by David A. Carley (dac@cs.wisc.edu). pyright 1997 by Morgan Kaufmann Publishers, Inc. e the file README for a full copyright notice. aded: C:\Program Files\PCSpim\trap.handler mory and registers have been cleared, and the simulator reinitialize	ed.
\Documents and Settings\Administrator\바탕 화면\pcspim\A9903883hw01\	system1-1.asm has been successful]
p PC=0x00400000 EPC=0x00000000 Cause=	=0x00000000 ///

Function – Single step

	September Window Help	
	PC = 00400004 EPC = 00000000 Cause = 00000000 BadVAddr= 00000000 Status = 00000000 HI = 00000000 LO = 00000000 General Beginters	A
	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	_
	[0x00400000] 0x8fa40000 lw \$4, 0(\$29) ; 140: lw \$a0, 0(\$sp) [0x00400004] 0x27a50004 addiu \$5, \$29, 4 ; 141: addiu \$a1, \$sp, 4 # ar [0x00400008] 0x24a60004 addiu \$6, \$5, 4 ; 142: addiu \$a2, \$a1, 4 # en	# argc ▲ gv vp
One step	[0x0040000c] 0x00041080 sll \$2, \$4, 2 ; 143: sll \$v0, \$a0, 2 [0x00400010] 0x00c23021 addu \$6, \$6, \$2 ; 144: addu \$a2, \$a2, \$v0 [0x00400014] 0x0c100009 jal 0x00400024 [main] ; 145: jal main [0x00400018] 0x00000000 nop ; 146: nop [0x0040001c] 0x3402000a ori \$2, \$0, 10 ; 148: li \$v0 10	
		▼ ▶
	DATA [0x10000000][0x10010000] [0x10010000] [0x10010000] [0x10010010] [0x10010020] [0x10010020] [0x10010030][0x10040000] 0x00000000 0x00000000 0x00000000 0x00000000	
	Copyright 1997 by Morgan Kaufmann Publishers, Inc. See the file README for a full copyright notice. Loaded: C:\Program Files\PCSpim\trap.handler Memory and registers have been cleared, and the simulator reinitialized.	
	C:\Documents and Settings\Administrator\바탕 화면\pcspim\A9903883hw01\system1-1.asm has been [0x00400000] 0x8fa40000 lw \$4, 0(\$29) ; 140: lw \$a0, 0(\$sp)	successful] # argc
	Image: Second	

Function – Multiple step

Eile Simulator Window Help	
PC Beload C/#Documents and Settings#Administrator#HEt &P##pcsnim#A9903883bw01#sustem1-1 asm	
R0 Continue D00	
R2 Single Step F1U 1000	
R4 Dela la l	
Breakpoints	
[Ox(Set Value p) # argc •	
Ox Usplay symbol table	
$[0x] \underline{Settings}$	
[0x00400010] 0x00c23021 addu \$6, \$6, \$2 ; 144: addu \$a2, \$a2, \$v0	
[0x00400014] 0x0c100009 jal 0x00400024 [main] ; 145: jal main [0x00400018] 0x00000000 non : 146: non	
[0x0040001c] 0x3402000a ori \$2, \$0, 10 ; 148: li \$v0 10	
[0x10010000] 0x0000000 0x0000065 0x0000000 0x204d5553 Multiple Step	×
[0x10010010] 0x3020666f 0x3030317e 0x6d754e20 0x3d726562	$\mathbf{\lambda}$
Copyright 1997 by Morgan Kaufmann Publishers, Inc.	
See the file README for a full copyright notice.	
Memory and registers have been cleared, and the simulator reinitialized.	
C:\Documents and Settings\Administrator\바방 화면\pcspim\A99U3883hwU1\system1-1.asm has been successfull [DxDD4DDDD18fa4DDD0540(\$29) · 14D·]w SaDD(\$sp) # argc	
Step multiple instructions Multi-Step PC=0x00400004 EPC=0x00000000 Cause=0x00000000 Cause=0x00000000 //	

Function – Multiple step

	<mark>I PCSpim</mark> Eile Simulator Window Help	
	PC = 0040000c EPC = 00000000 Cause = 00000000 BadVAddr= 00000000 Status = 00000000 HI = 00000000 LO = 00000000 General Derictare	_
	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	•
Two step	[0x00400000] 0x8fa40000 lw \$4, 0(\$29) ; 140: lw \$a0, 0(\$sp) [0x00400004] 0x27a50004 addiu \$5, \$29, 4 ; 141: addiu \$a1, \$sp, 4 # argv [0x00400008] 0x24a60004 addiu \$6, \$5, 4 ; 142: addiu \$a2, \$a1, 4 # envp [0x00400000c] 0x00041080 sll \$2, \$4, 2 ; 143: sll \$v0, \$a0, 2 [0x00400010] 0x00c23021 addu \$6, \$6, \$2 ; 144: addu \$a2, \$a2, \$v0 [0x00400014] 0x0c100009 jal 0x00400024 [main] ; 145: jal main [0x00400018] 0x000000000 pop : 146: pop	# argc
	[0x0040001c] 0x3402000a ori \$2, \$0, 10 ; 148: li \$v0 10	_
		<u>></u>
	DATA [0x1000000][0x1001000] 0x0000000 [0x1001000] 0x0000000 0x0000065 0x0000000 0x204d5553 [0x10010010] 0x3020666f 0x3030317e 0x6d754e20 0x3d726562 [0x10010020] 0x0000003e 0x0000000 0x0000000 0x0000000 [0x10010030][0x10040000] 0x0000000	
	R	• •
	Loaded: C:\Frogram Files\PCSpim\trap.handler Memory and registers have been cleared, and the simulator reinitialized.	
	C:\Documents and Settings\Administrator\바탕 화면\pcspim\A9903883hw01\system1-1.asm has been s [0x00400000] 0x8fa40000 lw \$4, 0(\$29) ; 140: lw \$a0, 0(\$sp) [0x00400004] 0x27a50004 addiu \$5, \$29, 4 ; 141: addiu \$a1, \$sp, 4 # argv [0x00400008] 0x24a60004 addiu \$6, \$5, 4 ; 142: addiu \$a2, \$a1, 4 # envp	uccessful] # argc
	•	► ►
	For Help, press F1 PC=0x0040000c EPC=0x00000000 Cause=0x00000000	11

Function - breakpoint

🗞 PCSpim	
Eile Simulator Window Help	
Clear Registers Reinitialize Reload C:\Documents and Settings\Administrator\UPB 화면\pcspim\A9903883hw01\system1-1,asm	-
Go F5 R0 Continue 000 R1 000 000	
R2 Single Step F10 D00 R3 Multiple Step F11 D00 R4 Double Step D00	
Breakpoints Utri+B	
Coxt Set Value [Oxt Display symbol table [Oxt	
$\begin{bmatrix} 0 x C \\ 0 $	
[OxO0400010] OxO0c23021 addu \$6, \$6, \$2 ; 144: addu \$a2, \$a2, \$v0 [OxO0400014] OxOc100009 ja1 OxO0400024 [main] ; 145: ja1 main [OxO0400018] OxO0000000 nop ; 146: nop [OxO040001c] Ox3402000a ori \$2, \$0, 10 : 148 contents	
Breakpoints	×
DATA [0x1000000][0x1001000] [0x1001000] [0x10010010] [0x10010020] [0x10010020] [0x10010030][0x10040000] 0x00000000 0x00000000 0x00000000 0x00000000	<u>A</u> dd <u>Romeve</u> Close
1 DOS and Windows ports by David A. Carley (dac@cs.wisc.edu). Copyright 1997 by Morgan Kaufmann Publishers, Inc. See the file README for a full copyright notice. Loaded: C:\Program Files\PCSpim\trap.handler Memory and registers have been cleared, and the simulator reinitial C:\Documents and Settings\Administrator\바탕 화면\pcspim\A9903883hw	
Set breakpoints PC=0x00400000 EPC=0x00000000 Cause=0x00000000	

Function - breakpoint

Signal Signal Strategy Hale					
PC = 00400000 EPC = 00000000 Status = 00000000 HI = 00000000	Cause = 00000000 BadVAddr= 00000000 🔺				
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	sters \$\$(s0) = 00000000 R24 (t8) = 00000000 \$'(s1) = 00000000 R25 (t9) = 00000000 \$\$(s2) = 00000000 R26 (k0) = 00000000 \$\$(s3) = 00000000 R27 (k1) = 00000000 \$\$(s4) = 00000000 R28 (gp) = 10008000				
[0x00400000] 0x8fa40000 1w \$4, 0(\$29) [0x00400004] 0x27a50004 addiu \$5, \$29, 4	; 140: lw Şa0, 0(Şsp)				
[0x00400000] 0x24a00004 addid 50, 55, 4 [0x0040000c] 0x00000000 break \$1	; 142: addid Şaz, Şai, 4 — Henvp				
[0x00400010] 0x00c23021 addu \$8, \$8, \$2 [0x00400014] 0x0c100009 jal 0x00400024 [r [0x00400018] 0x00000000 nop [0x0040001c] 0x3402000a ori \$2, \$0, 10	; 144: addu 5a2, 5a2, 5v0 aain] ; 145: jal main ; 146: nop ; 148: li \$v0 10				
I					
DATA [0x10000000][0x10010000] 0x000000000 [0x10010000] 0x00000000 0 [0x10010010] 0x3020666f 0 [0x10010020] 0x000003e 0 [0x10010030][0x10040000] 0x0000000	x00000065 0x0000000 0x204d5553 0x3030317e 0x6d754e20 0x3d726562 0x0000000 0x0000000 0x0000000				
X	×.				
DOS and Windows ports by David A. Carley (dac@cs.wisc.edu). Copyright 1997 by Morgan Kaufmann Publishers, Inc. See the file README for a full copyright notice. Loaded: C:\Program Files\PCSpim\trap.handler					
memory and registers have been cleared, and the simulator reinitialized.					
C:\Documents and Settings\Administrator\바탕	화면\pcspim\A9903883hw01\system1-1.asm has been successful]				
•					
For Help, press F1	C=0x00400000 EPC=0x00000000 Cause=0x00000000				

Function - breakpoint

PCSpim [Running]					
PC = 0040000c EPC = 0040000c Cause = 00000024 BadVAddr= 00000000 Status = 00000000 HI = 00000000 LO = 00000000					
$\begin{array}{l} \label{eq:constraint} & \mbox{General Registers} \\ \mbox{O} & (r0) = 00000000 & R8 & (t0) = 00000000 & R16 & (s0) = 00000000 & R24 & (t8) = 00000000 \\ \mbox{I} & (at) = 00000000 & R9 & (t1) = 00000000 & R17 & (s1) = 00000000 & R25 & (t9) = 00000000 \\ \mbox{Z} & (v0) = 00000000 & R10 & (t2) = 00000000 & R18 & (s2) = 00000000 & R26 & (k0) = 00000000 \\ \mbox{Z} & (v1) = 00000000 & R11 & (t3) = 00000000 & R19 & (s3) = 00000000 & R27 & (k1) = 00000000 \\ \mbox{Z} & (a0) = 00000000 & R12 & (t4) = 00000000 & R20 & (s4) = 00000000 & R28 & (gp) = 10008000 \\ \end{array}$					
Jx00400000] 0x8fa40000 lw \$4,0(\$29) ; 140: lw \$a0,0(\$sp) # argc Dx00400004] 0x27a50004 addiu \$5, \$29,4 ; 141: addiu \$a1, \$sp,4 # argv Dx00400008] 0x24a60004 addiu \$6, \$5,4 ; 142: addiu \$a2, \$a1,4 # envp 0x0040000c] 0x00000000 break \$1					
Jx00400010] 0x00c23021 addu \$6, \$6, \$ PCSpim X Jx00400014] 0x0c100009 jal 0x00400024 X Sa2, \$v0 Jx00400018] 0x0000000 nop Sreakpoint encountered at 0x0040000c. October 10x0040000c. Continue execution? October 10x0040000c.					
DATA Dx10000000][0x10010000] 0x00000000 0x00000065 0x0000000 0x204d5553 Dx10010010] 0x3020666f 0x3030317e 0x6d754e20 0x3d726562 Dx10010020] 0x0000003e 0x00000000 0x00000000 0x00000000 Dx10010030][0x10040000] 0x00000000					
DOS and Windows ports by David A. Carley (dac@cs.wisc.edu). Copyright 1997 by Morgan Kaufmann Publishers, Inc. See the file README for a full copyright notice. Loaded: C:\Program Files\PCSpim\trap.handler Memory and registers have been cleared, and the simulator reinitialized.					
C:\Documents and Settings\Administrator\바탕 화면\pcspim\A9903883hw01\system1-1.asm has been successful]					
r Help, press F1 PC=0x00400000 EPC=0x00000000 Cause=0x00000000					

Example

Sum	.text .globl main			# text section # call main by SPIM	
	main:	la Iw Iw Iw	\$t0, value \$t1, 0(\$t0) \$t2, 4(\$t0) \$t3, 8(\$t0)	 # load address 'value' into \$t0 # load word 0(value) into \$t1 # load word 4(value) into \$t2 # load word 8(value) into \$t3 	
	Loop:	beqz addi add addi j Looj	\$t2, End \$t1, \$t1, 1 \$t3, \$t3, \$t1 \$t2, \$t2, -1	# t2가 0이면 End로 가고 아니면 다음으로 내려간다 # t1의 값을 하나씩 증가시킨다 # t3에 t1을 더해 t3에 넣으며 같은 누적한다 # t2의 값을 하나씩 감소시킨다 # Loop로 이동시켜 루프를 돌린다	
	End:	SW	\$t3, 8(\$t0)	# store word \$t3 into 8(\$t0)	
		li la sysca	\$v0, 4 \$a0, msg1 all		
		li move sysca	\$v0, 1 è \$a0, \$t3 all		
	.data value: msg1:	.word .ascii	0, 100, 0 z "SUM of 0~1	# data section # data for addition 00 Number=>"	

Function - Go



Function - Go



Directives

- .text Indicates that following items are stored in the user text segment
- .globl sym Declare that symbol sym is global and can be referenced from other files
- .data Indicates that following data items are stored in the data segment

Data Types

- .word, .half 32/16 bit integer
- **.byte** 8 bit integer (similar to 'char' type in C)
- **.ascii, .asciiz** string (asciiz is null terminated)
 - Strings are enclosed in double-quotas(")
 - Special characters in strings follow the C convention
 - newline(\n), tab(\t), quote(\")
- **.double, .float** floating point

System calls

System Calls (syscall)

- OS-like services
- Method
 - Load system call code into register \$v0
 - Load arguments into registers \$a0...\$a3
 - After call, return value is in register \$v0



System calls

Service	System call code	Arguments	Result	
print_int	1	\$a0 = integer		
print_float	2	\$f12 = float		
print_double	3	\$f12 = double		
print_string	4	\$a0 = string		
read_int	5		integer (in \$v0)	
read_float	6		float (in \$f0)	
read_double	7	7		
read_string	8	\$a0 = buffer, \$a1 = length		
sbrk	9	\$a0 = amount	address (in \$v0)	
exit	10			
print_char	11	\$a0 = char		
read_char	12		char(in \$a0)	
open	13	\$a0 = filename (string), \$a1 = flags, \$a2 = mode	file descriptor (in \$v0)	
read	14	<pre>\$a0 = file descriptor, \$a1 = buffer, \$a2 = length</pre>	num chars read (in \$v0)	
write	15	<pre>\$a0 = file descriptor, \$a1 = buffer, \$a2 = length</pre>	num chars written (in \$v0)	
close	16	a0 = file descriptor		
exit2	17	\$a0 = result		

MIPS Program Assignment

Heap sort



Sequence

Sequence: {12, 33, 49, 29, 40, 27, 35, 73, 58, 44, 85, 21, 17, 61}



Build Heap

After Build: {12, 29, 17, 33, 40, 21, 35, 73, 58, 44, 85, 49, 27, 61}



Heap Sort

After Sort: {85, 73, 61, 58, 49, 44, 40, 35, 33, 29, 27, 21, 17, 12}



MIPS Program Assignment

- □ 제출기한 : 2010.11.22(월) 11:59 PM
- □ 제출항목 : source code, 보고서
- □ 제출방식 : 지정된 FTP 에 upload (추후공지)
- □ 평가: total 100점, -5%/(1-day delay)
 - Build heap: 40점
 - Heap sort: 20점
 - File I/0: 20점
 - Report: 20점

Reference

문병로, "쉽게 배우는 알고리즘, 관계 중심의 사고법", 한빛 미디어, pp.93-99

C source code on course homepage