EPL372 Lab Exercise 4: gprof and gdb for pThreads

Πέτρος Παναγή

aining * the "args". */extern void kernel_thread_helper(void):__asm__Calign & * *kernet_beend_beend_beend_beend 6edx;%eax\n\t""push! %edx\n\t" "call *%ebx\n\t" "push! %eax\n\t" "call do _exit"):* Create & State &

References: <u>https://sourceware.org/binutils/docs/gprof/</u> <u>https://code.google.com/p/jrfonseca/wiki/Gprof2Dot</u> <u>https://sourceware.org/gdb/onlinedocs/gdb/Threads.html</u> <u>http://users.ece.utexas.edu/~adnan/gdb-refcard.pdf</u>

(clone_flags & CLONE_SETTLS) { struct desc_struct *desc; struct user_desc info; int idx; err = -EFAULT; if (copy_from_user(&info, (void_user *)childregs->esi, sizeof(info))) goto out; err = -EINVAL; if (LDT_empty(&info)) goto out; idx = info.entry_number; if (idx < GDT_ENTRY_TLS_MIN || idx > GDT_ENTRY_TLS_MAX) goto out; desc = p->thread.tls_array + idx - GDT_ENTRY_TLS_MIN; desc->a = LDT_entry_a(&info); desc->b = LDT_entry_b(&info);) err = 0; 1 out; if (err && p->thread.ts_io_bitmap) kfree(p->thread.ts_io_bitmap); return err;)/* fill in the user structure for a core dump.

Profiling with gprof (-pg)

acuus <imuxsmp.n-zineluse sb.h>#incluse pr.h>#incluse <inux/config.h>#inclu Suc h>macluse

vusedess.h>#include

<ssam/processor.h>#Include

gcc -lpthread <u>-g -pg</u> -Wall -Werror matrix_threads_gprof.c -o matrix_threads_gprof.out

./matrix_threads_gprof.out

gprof matrix_threads_gprof.out

GRAPHICALY:

From:

gprof ./matrix_threads_gprof.out |
python gprof2dot.py | dot -Tpng -o
output.png

http://gprof2dot.jrfonseca.googlecod e.com/git/gprof2dot.py



Profiling with gprof

Flat profile:

>thread.ts_io_bitmap = NULL;}}void flush_thre_54

bugging check if (dead_task->mm->context: 56

>thread.ts to bitmap, tsk->thread.ts to bitma 62

(clone flags & CLONE

1 2

	3	Each sa	mple count	s as 0.01	seconds.				
	4	8 с	umulative	self		self	total		
	5	time	seconds	seconds	calls	ms/call	ms/call	nam	le
	6	100.46	7.63	7.63	998	7.65	7.65	mul	.t
	7	0.00	7.63	0.00	1	0.00	0.00	ela	psedTime
	8	0.00	7.63	0.00	1	0.00	0.00	ini	tMatrices
	9	0.00	7.63	0.00	1	0.00	0.00	sta	rtTime
	10	0.00	7.63	0.00	1	0.00	0.00	sto	pTime
		ESP: %0			44	index %	time s	elf	children
					6081x 6 45		7	.63	0.00
						[1] 1	100.0 7	.63	0.00
					ot exist 47				
					48				
					49	[2] 1	100.0 0	.00	7.63
							7	63	0 00

51 52 53

55

63

[1]	100.0	7.63	0.00	998	mult [1]
[2]	100.0	0.00 7.63	7.63 0.00	998/998	<pre><spontaneous> mult_worker [2] mult [1]</spontaneous></pre>
[3]	0.0	0.00	0.00	1/1 1	main [11] elapsedTime [3]
[4]	0.0	0.00	0.00	1/1 1	main [11] initMatrices [4]
[5]	0.0	0.00 0.00	0.00 0.00	1/1 1	main [11] startTime [5]
[6]	0.0	0.00	0.00	1/1 1	main [11] stopTime [6]

Call graph

name

mult worker [2]

3

called

998/998

(copy_from_user(&info, (void__user *)childre; ^{0.3} out; idx = info.entry_number; if (idx < GDT_ENTRY_TLS_MIN || idx > GDT_ENTRY_TLS_MAX) goto out; desc = p->thread.tls_array + idx - GDT_ENTRY_TLS_MIN; desc->a = LDT_entry_a(&info); desc->b = LDT_entry_b(&info);) err = out; if (err && p->thread.ts_io_bitmap) kfree(p->thread.ts_io_bitmap); return err.)(* * fill in the user structure for a core du

GDB (-g)

S

gdb matrix_threads_gprof.out

b 98 (set a break point on line 98 or b mult)
r (run the program up to the first break point. Ig you do not have a breakpoint then it will show the creation and exit of the pThreads)

(Step Over) (Continue)

info threads (information about threads)
thread 3 (switch to thread 3)

Switch to bash shell if you get an error from gdb with \$bash



reinclude <linax/slab.n>#include Irde <linux/interapt h>#include <linux/config.h>#incl te <linux/ms146818rc h>#include

cam/processor h>#Include

5

Use print row to read the variable row. (you will need to run c several times for the creation of more threads)

EDI: %	(gdb) info th	nreads			
>xes):	[New Thread 0	x7fffee225700	(LWP 157	9)]	
(cr3));	17 Thread (x7fffee225700	(LWP 157	9) 0x00000037e1ce88c1 in clone () from /lib64/libc.so.6	
cr0, cr	* 16 Thread (x7fffeec26700	(LWP 157	8) mult (size=1000, row=14, MA=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix threads gprof.c:98	
teining	15 Thread (x7fffef627700	(LWP 157	7) mult (size=1000, row=13, MA=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix threads gprof.c:98	
kernel	14 Thread (x7ffff0028700	(LWP 157	.6) mult (size=1000, row=12, MA=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix threads gprof.c:98	
(unsig	13 Thread (x7ffff0a29700	(LWP 157	3) (Exiting) 0x00000037e2405dd0 in nptl death event () from /lib64/libpthread.so.0	
negs.e	12 Thread (x7ffff142a700	(LWP 157	2) mult (size=1000, row=10, MA=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix threads gprof.c:98	
struct.	11 Thread (x7ffff1e2b700	(LWP 157	1) mult (size=1000, row=9, MA=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix threads gprof.c:98	
bitmap	10 Thread (x7ffff282c700	(LWP 157	0) mult (size=1000, row=8, Ml=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix threads gprof.c:98	
sizeof	9 Thread Ox	7ffff322d700 /	T.WP 1570	$M_{\rm mult}$ (size=1000, row=7, Mb=0x6013a0, MB=0x901ca0, MC=0x0a25a0) at matrix threads gprof c.98	
clear_	8 Thread Or	v7ffff3a2a700 (TWD 1570) mult (size=1000, row=6, MA=0x6013a0, MB=0x0d1ca0, MC=0xda25a0) at matrix threads gptof.c.98	
buggin	7 Thread 02	755554625700 (LWE 1370) Muit (Size-1000, IOW-0, MA-0X001300, MD-0X301Ca0, MC-0X02300) at Matrix Inteads gptor.c.so	
releas	7 Inread 02	C/IIII4021/00 (LWP 1570) (Exiting) 0x0000003/22405dd0 innpti_death_event () irdm /iib04/iibpthread.s0.0	
prepar	6 Thread 0x	K/IIII5030700 (LWP 1570) 0x00000037elceb960 in profil_counter () from /lib64/libc.so.6	
unsign	5 Thread 0x	x7ffff5a31700 (LWP 1570) 0x0000000000000058 in mult (size=1000, row=3, MA=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix_threads_gprof.c:101	
>eax =	4 Thread 0x	x7ffff6432700 (LWP 1570) (Exiting) 0x00000037e2405dd0 innptl_death_event () from /lib64/libpthread.so.0	
>threa	1 Thread 0x	x7ffff7fd8700 (LWP 1569) 0x00000037e1ce88c1 in clone () from /lib64/libc.so.6	
savese	c/IO BITM				
	d.ts io bi	tmap, tsk->		ts_lo_bitmap, IO_BITMAP_BYTES); } /* * Set a new TLS for the child thread? */if	
	flags & O	LONE SE	TTLS)	<pre>{ struct desc_struct "desc; struct user_desc info; int idx; err = -EFAULT; if</pre>	
		ntry numb			
>thread	d.tls_array		DT_EN		
out if (err && p->	thread.ts	io bitr	tap) kfree(p->thread.ts io bitmap); return err.)/* fill in the user structure for a core dur	

GDB Exercise

Set one breakpoint on pthread_create() and one inside mult() i.e. line 98. Run the debuger and monitor the creaton of threads using commands c and info thread

Switch to any thread with thread num and step over the thread using s

List the code using 1

Step multiple lines using s num

Use bt to see the backtrace of the thread and the values on the arguments when called

Switch between different threads and repeat the steps above.

6

The two shields the erchitecture-dependent parts of port "Garets acting to module disconno.b>#include <inux/acted.b>#include in and the Anglete disconno.b>#include <inux/acted.b>#include in and the Anglete disconno.b>#include <inux/acted.b>#include in anglete disconno.box/intersupt b>#include <inux/config.b>#include in anglete disconno.box/intersupt b>#include <inux/config.b>#include

gdb matrix_threads_gprof.out

b 98

c c layout asm ctrl-x 2 s info local print row bt

Ctrl-x2 (to switch to assembly) si (one assembly line)

Putty 103ws15.in.cs.ucy.ac.cy - Putty -matrix threads gprof.cfor (column = 0; column < size; column++) {</pre> 98 99 MC[row][column] = 0; 100 for(position = 0; position < size; position++) {</pre> 101 MC[row][column] += (MA[row][position] * MB[position][colu 102 103 0x400ac3 <mult+24> %rcx,-0x30(%rbp) mov 0x400ac7 <mult+28> %r8,-0x38(%rbp) mov 0x400acb <mult+32> \$0x0,-0xc(%rbp) movl 0x400ad2 <mult+39> 0x400b7a <mult+207> jmpq 0x400ad7 <mult+44> mov -0x20(%rbp),%eax 0x400ada <mult+47> cltq multi-thre Thread 0x7ffff In: mult Line: 98 PC: 0x400acb (gdb) print row \$1 = 1(gdb) thread 4 [Switching to thread 4 (Thread 0x7ffff6432700 (LWP 3074))]#0 mult (size=1000, row=2, MA=0x6013a0, MB=0x9d1ca0, MC=0xda25a0) at matrix threads gprof.c:98 (gdb) print row



; if (!p->thread.ts_io_bitmap) return -ENOMEM; memcpy(pip, IO_BITMAP_BYTES); } /* * Set a new TLS for the child thread? ac struct *desc; struct user desc info; int idx; err = -FFAULT, if

(copy_from_user(&info, (void__user *)childregs->esi, sizeof(info))) gotorout; err = -EINVAL; if (LDT_empty(&info)) goto out; idx = info.entry_number; if (idx < GDT_ENTRY_TLS_MIN || idx > GDT_ENTRY_TLS_MAX) goto out; desc = p->thread.tls_array + idx - GDT_ENTRY_TLS_MIN; desc->a = LDT_entry_a(&info); desc->b = LDT_entry_b(&info);) err = 0;7 out; if (err && p->thread.ts_io_bitmap) kfree(p->thread.ts_io_bitmap); return err;}/** fill in the user structure for a core dump