Steven Tang

CSE 460

Lab 6

Pthread\_demo.cpp(MODDED):

1. /\*
2. pthreads\_demo.cpp
3. A very simple example demonstrating the usage of pthreads.
4. Compile: g++ -o pthreads\_demo pthreads\_demo.cpp -lpthread
5. Execute: ./pthreads\_demo
6. \*/
8. #include <pthread.h>
9. #include <stdio.h>
11. **using** **namespace** std;
13. //The thread
14. **void** \*runner ( **void** \*data )
15. {
16. **char** \*tname = ( **char** \* )data;
18. printf("I am %s\n", tname );
20. pthread\_exit ( 0 );
21. }
23. **int** main ()
24. {
25. pthread\_t id1, id2, id3;           //thread identifiers
26. pthread\_attr\_t attr1, attr2, attr3;  //set of thread attributes
27. **char** \*tnames[3] = { "Thread 1 likes rice", "Thread 2 likes burgers", "Thread 3 likes tacos" }; //names of threads
28. //get the default attributes
29. pthread\_attr\_init ( &attr1 );
30. pthread\_attr\_init ( &attr2 );
31. pthread\_attr\_init ( &attr3 );
33. //create the threads
34. pthread\_create ( &id1, &attr1, runner, tnames[0] );
35. pthread\_create ( &id2, &attr2, runner, tnames[1] );
36. pthread\_create ( &id3, &attr3, runner, tnames[2] );
38. //wait for the threads to exit
39. pthread\_join ( id1, NULL );
40. pthread\_join ( id2, NULL );
41. pthread\_join ( id3, NULL );
43. **return** 0;
44. }

pthread typescript:

1. Script started on Thu 16 Feb 2017 12:27:53 AM PST
2. [004532176@jb359-16 lab6]$ g++ -o pthread\_demo pthread\_demo.cpp -lpthread
3. pthread\_demo.cpp: In function ‘**int** main()’:
4. pthread\_demo.cpp:27:95: warning: **deprecated** conversion from string constant to ‘**char**\*’ [-Wwrite-strings]
5. **char** \*tnames[3] = { "Thread 1 likes rice", "Thread 2 likes burgers", "Thread 3 likes tacos" }; //names of threads
6. ^
7. pthread\_demo.cpp:27:95: warning: **deprecated** conversion from string constant to ‘**char**\*’ [-Wwrite-strings]
8. pthread\_demo.cpp:27:95: warning: **deprecated** conversion from string constant to ‘**char**\*’ [-Wwrite-strings]
9. [004532176@jb359-16 lab6]$ ./pthread\_demo
10. I am Thread 1 likes rice
11. I am Thread 2 likes burgers
12. I am Thread 3 likes tacos
13. [004532176@jb359-16 lab6]$ exit
14. exit
16. Script done on Thu 16 Feb 2017 12:28:18 AM PST

SDLthread\_demo.cpp(MODDED)

1. /\*
2. sdlthread\_demo.cpp
3. A very simple example demonstrating the usage of sdl threads.
4. Compile:  g++ -o sdlthread\_demo sdlthread\_demo.cpp -lSDL -lpthread
5. Execute:  ./sdlthread\_demo
6. \*/
8. #include <SDL/SDL.h>
9. #include <SDL/SDL\_thread.h>
10. #include <stdio.h>
11. #include <iostream>
12. **using** **namespace** std;
14. //The thread
15. **int** runner ( **void** \*data )
16. {
17. **char** \*tname = ( **char** \* )data;
19. printf("I am %s\n", tname );
20. **return** 0;
21. }
23. **int** main ()
24. {
25. SDL\_Thread \*id1, \*id2, \*id3;                //thread identifiers
26. **char** \*tnames[3] = { "Thread 1, who are you?", "Thread 2, I am you", "Thread 3, I'm both of you" }; //names of threads

29. //create the threads
30. id1 = SDL\_CreateThread ( runner, tnames[0] );
31. id2 = SDL\_CreateThread ( runner, tnames[1] );
32. id2 = SDL\_CreateThread ( runner, tnames[2] );
34. //wait for the threads to exit
35. SDL\_WaitThread ( id1, NULL );
36. SDL\_WaitThread ( id2, NULL );
37. SDL\_WaitThread ( id3, NULL );
39. **return** 0;
40. }

SDLTHREAD Typescript:

1. Script started on Thu 16 Feb 2017 12:23:35 AM PST
2. [004532176@jb359-16 lab6]$ g++ -o sdlthread\_demo sdlthread\_demo.cpp -lSDL -lpthread
3. sdlthread\_demo.cpp: In function ‘**int** main()’:
4. sdlthread\_demo.cpp:26:99: warning: **deprecated** conversion from string constant to ‘**char**\*’ [-Wwrite-strings]
5. **char** \*tnames[3] = { "Thread 1, who are you?", "Thread 2, I am you", "Thread 3, I'm both of you" }; //names of threads
6. ^
7. sdlthread\_demo.cpp:26:99: warning: **deprecated** conversion from string constant to ‘**char**\*’ [-Wwrite-strings]
8. sdlthread\_demo.cpp:26:99: warning: **deprecated** conversion from string constant to ‘**char**\*’ [-Wwrite-strings]
9. [004532176@jb359-16 lab6]$ ./sdlthread\_demo
10. I am Thread 1, who are you?
11. I am Thread 2, I am you
12. I am Thread 3, I'm both of you
13. Segmentation fault (core dumped)
14. [004532176@jb359-16 lab6]$ exit
15. exit
17. Script done on Thu 16 Feb 2017 12:24:03 AM PST

3. UNIX SEMAPHORE

Just reviewed semaphore functions

4. USING SEMAPHORES

Sema1\_1 typescript

1. Script started on Thu 16 Feb 2017 12:35:21 AM PST
2. [004532176@jb359-16 lab6]$ g++ -o sema1 sema1.cpp
3. [004532176@jb359-16 lab6]$ ./sema1 &
4. [1] 1646
5. [004532176@jb359-16 lab6]$ elelelelelelelelelel
6. 1646 finished!
7. ^C
8. [1]+  Done                    ./sema1
9. [004532176@jb359-16 lab6]$ exit
10. exit
12. Script done on Thu 16 Feb 2017 12:35:58 AM PST

When I ran ./sema1& I saw it print out “e” and “l” after I ran the program. This goes in and out of the critical section and is printed in lowercase.

Sema1\_a typescript

1. Script started on Thu 16 Feb 2017 12:37:26 AM PST
2. [004532176@jb359-16 lab6]$ g++ -o sema1 sema1.cpp
3. [004532176@jb359-16 lab6]$ ./sema1 a
4. ELELELELELELELELELEL
5. 1818 finished!
6. [004532176@jb359-16 lab6]$ exit
7. exit
9. Script done on Thu 16 Feb 2017 12:38:12 AM PST

When I ran ./sema1 a I saw that “e” and “l” would print out 10 times after we ran the command. This occurs because it goes in and out of the critical section that much time. Also it’s printed out in caps.

Sema1 MODDED

1. //sema1.cpp
2. //MODDED
3. #include <unistd.h>
4. #include <stdlib.h>
5. #include <stdio.h>
6. #include <sys/types.h>
7. #include <sys/ipc.h>
8. #include <sys/sem.h>
9. #include <iostream>
10. #include <stdio.h>
11. #include <string.h>
12. **using** **namespace** std;
14. **static** **int** sem\_id;      //semaphore id
16. #if defined(\_\_GNU\_LIBRARY\_\_) && !defined(\_SEM\_SEMUN\_UNDEFINED)
17. /\* union semun is defined by including <sys/sem.h> \*/
18. #else
19. /\* according to X/OPEN we have to define it ourselves \*/
20. **union** semun {
21. **int** val;                  /\* value for SETVAL \*/
22. **struct** semid\_ds \*buf;     /\* buffer for IPC\_STAT, IPC\_SET \*/
23. unsigned **short** \*array;    /\* array for GETALL, SETALL \*/
24. /\* Linux specific part: \*/
25. **struct** seminfo \*\_\_buf;    /\* buffer for IPC\_INFO \*/
26. };
27. #endif
29. //initializes semaphore using SETVAL
30. **static** **int** set\_semvalue ( **int** val )
31. {
32. **union** semun sem\_union;// sem\_union;
34. sem\_union.val = val;
35. **if** ( semctl ( sem\_id, 0, SETVAL, sem\_union ) == -1 ) **return** ( 0 );
36. **return** 1;
37. }
39. //delete semaphore
40. **static** **int** del\_semvalue ()
41. {
42. **union** semun sem\_union;// sem\_union;
44. sem\_union.val = 1;
45. **if** ( semctl ( sem\_id, 0, IPC\_RMID, sem\_union ) == -1 ) **return** ( 0 );
46. **return** 1;
47. }
49. **static** **int** SEM\_DOWN ()
50. {
51. **struct** sembuf b;
53. b.sem\_num = 0;
54. b.sem\_op = -1;    //P(), i.e. down()
55. b.sem\_flg = SEM\_UNDO;
56. **if** ( semop ( sem\_id, &b, 1 ) == -1 ) {
57. cout << "Semaphore DOWN() failed!" << endl;
58. **return** 0;
59. }
61. **return** 1;
62. }
64. **static** **int** SEM\_UP()
65. {
66. **struct** sembuf b;
68. b.sem\_num = 0;
69. b.sem\_op =  1;        //V(), i.e. UP()
70. b.sem\_flg = SEM\_UNDO;
71. **if** ( semop ( sem\_id, &b, 1 ) == -1 ) {
72. cout << "Semaphore UP() failed!" << endl;
73. **return** 0;
74. }
75. **return** 1;
76. }
77. **int** main ( **int** argc, **char** \*argv[] )
78. {
79. **int** i, pause\_time;
80. **char** ce = 'e', cl = 'l';
82. srand ( ( unsigned **int** ) getpid() );  //seed RNG with process id
84. sem\_id  = semget ( (key\_t) 1234, 1, 0666 | IPC\_CREAT );
86. **if** ( argc > 0 ) {
87. **if** ( !set\_semvalue( 1 ) ) {     //process can enter CS
88. cout << "Semaphore initialized failed!" << endl;
89. exit ( EXIT\_FAILURE );
90. }
91. **if** ( argc > 1 ) {
92. ce = 'E';
93. cl = 'L';
94. }
95. sleep ( 1 );
96. } **else** {
97. **if** ( !set\_semvalue( 0 ) ) {     //process will be blocked initially
98. cout << "Semaphore initialized failed!" << endl;
99. exit ( EXIT\_FAILURE );
100. }
101. sleep ( 1 );
102. }
104. //enter and leave critical section 10 times
105. **if** (strcmp(argv[1],"1") == 0){
106. **for** ( i = 0; i < 10; i++ ){
107. **if** ( !SEM\_DOWN () ) exit ( EXIT\_FAILURE );
108. cout << ce; fflush ( stdout );    //entering critical section
109. pause\_time = rand() % 3;        //simulate critical section
110. sleep ( pause\_time );
111. cout << cl; fflush ( stdout );    //leaving critical section
112. **if** ( !SEM\_UP() ) exit ( EXIT\_FAILURE ); //signal other waiting process
113. pause\_time = rand() % 2;
114. sleep ( pause\_time );
115. }
116. cout << endl << getpid() << " finished!" << endl;
117. **if** ( argc > 0 ) {
118. sleep ( 2 );
119. del\_semvalue ();
120. }
121. }
122. **else** **if**(strcmp(argv[1],"0") == 0){
123. cout<<"nothing" << endl;
124. sleep(999999);
125. }
126. exit ( EXIT\_SUCCESS );
127. }

XV6 Scheduling

1. [004532176@jb359-16 xv6]$ make qemu-nox
2. which: no qemu in (/usr/local/bin:/opt/eclipse:/opt/scilab/bin:/opt/android-studio/bin:/opt/argouml:/usr/lib64/openmpi/bin:/usr/local/cuda/bin:/share/bin:/opt/Xilinx/14.7/ISE\_DS/ISE/bin/lin64:/opt/Xilinx/14.7/ISE\_DS/common/bin/lin64:/opt/android-sdk-linux/tools:/opt/android-sdk-linux/platform-tools:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/u/cse/004532176/bin/)
3. qemu-system-i386 -nographic -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
5. (process:10443): GLib-WARNING \*\*: gmem.c:482: custom memory allocation vtable not supported
6. xv6...
7. cpu1: starting
8. cpu0: starting
9. Process initcode with pid 1 running
10. Process initcode with pid 1 running
11. sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
12. Process initcode with pid 1 running
13. Process initcode with pid 1 running
14. Process initcode with pid 1 running
15. Process initcode with pid 1 running
16. Process initcode with pid 1 running
17. Process initcode with pid 1 running
18. Process initcode with pid 1 running
19. Process initcode with pid 1 running
20. Process initcode with pid 1 running
21. Process initcode with pid 1 running
22. Process init with pid 1 running
23. Process init with pid 1 running
24. Process init with pid 1 running
25. Process init with pid 1 running
26. Process init with pid 1 running
27. Process init with pid 1 running
28. Process init with pid 1 running
29. Process init with pid 1 running
30. Process init with pid 1 running
31. Process init with pid 1 running
32. Process init with pid 1 running
33. inProcess init with pid 1 running
34. it: starting sh
35. Process init with pid 1 running
36. Process init with pid 2 running
37. Process init with pid 2 running
38. Process init with pid 2 running
39. Process init with pid 2 running
40. Process init with pid 2 running
41. Process init with pid 2 running
42. Process init with pid 2 running
43. Process init with pid 2 running
44. Process init with pid 2 running
45. Process init with pid 2 running
46. Process init with pid 2 running
47. Process init with pid 2 running
48. Process init with pid 2 running
49. Process init with pid 2 running
50. Process init with pid 2 running
51. Process init with pid 2 running
52. $ foo 4
53. Process sh with pid 2 running
54. Process sh with pid 2 running
55. Process sh with pid 3 running
56. exec foo failed
57. Process sh with pid 2 running
58. $ foo 4
59. Process sh with pid 2 running
60. Process sh with pid 4 running
61. eProcess sh with pid 4 running
62. xec foo failed
63. Process sh with pid 2 running
64. $ q
65. Process sh with pid 2 running
66. Process sh with pid 5 running
67. Process sh with pid 5 running
68. exec q failed
69. Process sh with pid 2 running
70. $ quit
71. Process sh with pid 2 running
72. Process sh with pid 6 running
73. Process sh with pid 6 running
74. exec quit failed
75. Process sh with pid 2 running
76. $
77. Process sh with pid 2 running
78. Process sh with pid 7 running
79. Process sh with pid 7 running
80. exec  failed
81. Process sh with pid 2 running

Changing proc.c

1. [004532176@jb359-16 xv6]$ make qemu-nox
2. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
3. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
4. -c -o proc.o proc.c
5. ld -m elf\_i386 -T kernel.ld -o kernel entry.o bio.o console.o
6. exec.o file.o fs.o ide.o ioapic.o kalloc.o kbd.o lapic.o log.o
7. main.o mp.o picirq.o pipe.o proc.o sleeplock.o spinlock.o
8. string.o swtch.o syscall.o sysfile.o sysproc.o timer.o trapasm.o
9. trap.o uart.o vectors.o vm.o -b binary initcode entryother
10. objdump -S kernel > kernel.asm
11. objdump -t kernel | sed '1,/SYMBOL TABLE/d; s/ .\* / /; /^$/d' >
12. kernel.sym
13. dd **if**=/dev/zero of=xv6.img count=10000
14. 10000+0 records in
15. 10000+0 records out
16. 5120000 bytes (5.1 MB) copied, 0.116467 s, 44.0 MB/s
17. dd **if**=bootblock of=xv6.img conv=notrunc
18. 1+0 records in
19. 1+0 records out
20. 512 bytes (512 B) copied, 0.00122226 s, 419 kB/s
21. dd **if**=kernel of=xv6.img seek=1 conv=notrunc
22. 357+1 records in
23. 357+1 records out
24. 183072 bytes (183 kB) copied, 0.00416878 s, 43.9 MB/s
25. which: no qemu in
26. (/usr/local/bin:/opt/eclipse:/opt/scilab/bin:/opt/androidstudio/bin:/opt/argouml:/usr/lib64/openmpi/bin:/usr/local/cuda/b
27. in:/share/bin:/opt/Xilinx/14.7/ISE\_DS/ISE/bin/lin64:/opt/Xilinx/
28. 14.7/ISE\_DS/common/bin/lin64:/opt/android-sdklinux/tools:/opt/android-sdk-linux/platformtools:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/u/cse/0
29. 04867222/bin/.)
30. qemu-system-i386 -nographic -drive
31. file=fs.img,index=1,media=disk,format=raw -drive
32. file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
33. (process:26800): GLib-WARNING \*\*: gmem.c:482: custom memory
34. allocation vtable not supported
35. xv6...
36. cpu1: starting
37. cpu0: starting
38. Process initcode with pid 1 running
39. Process initcode with pid 1 running
40. Process initcode with pid 1 running
41. sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2
42. inodestart 32 bmap start 58
43. Process initcode with pid 1 running
44. Process initcode with pid 1 running
45. Process initcode with pid 1 running
46. Process initcode with pid 1 running
47. Process initcode with pid 1 running
48. Process initcode with pid 1 running
49. Process initcode with pid 1 running
50. Process initcode with pid 1 running
51. Process initcode with pid 1 running
52. Process initcode with pid 1 running
53. Process init with pid 1 running
54. init: starting sh
55. Process init with pid 2 running
56. Process init with pid 2 running
57. Process init with pid 2 running
58. Process init with pid 2 running
59. Process init with pid 2 running
60. Process init with pid 2 running
61. Process init with pid 2 running
62. Process init with pid 2 running
63. Process init with pid 2 running
64. Process init with pid 2 running
65. Process init with pid 2 running
66. Process init with pid 2 running
67. Process init with pid 2 running
68. Process init with pid 2 running
69. Process init with pid 2 running
70. Process init with pid 2 running
71. $ foo
72. Process sh with pid 2 running
73. Process sh with pid 3 running
74. Process sh with pid 3 running
75. Process sh with pid 3 running
76. Process sh with pid 3 running
77. Process sh with pid 3 running
78. Process sh with pid 3 running
79. Process sh with pid 3 running
80. Process sh with pid 3 running
81. Process foo with pid 3 running
82. Parent 3 creating child 4
83. Process foo with pid 4 running
84. Process foo with pid 4 running
85. Child 4 created
86. Process foo with pid 3 running
87. Process foo with pid 4 running
88. Process foo with pid 3 running
89. Process foo with pid 4 running
90. Process foo with pid 3 running
91. Process foo with pid 4 running
92. Process foo with pid 3 running
93. Process foo with pid 4 running
94. Process foo with pid 3 running
95. Process foo with pid 4 running
96. Process foo with pid 3 running
97. Process foo with pid 4 running
98. Process foo with pid 3 running
99. Process foo with pid 4 running
100. Process foo with pid 3 running
101. Process foo with pid 4 running
102. Process foo with pid 3 running
103. Process foo with pid 4 running
104. Process foo with pid 3 running
105. Process foo with pid 4 running
106. Process foo with pid 3 running
107. Process foo with pid 4 running
108. Process foo with pid 3 running
109. Process foo with pid 4 running
110. Process foo with pid 3 running
111. Process foo with pid 4 running
112. Process foo with pid 3 running
113. Process foo with pid 4 running
114. Process foo with pid 3 running
115. Process foo with pid 4 running
116. Process foo with pid 3 running
117. Process foo with pid 4 running
118. Process foo with pid 3 running
119. Process foo with pid 4 running
120. Process foo with pid 3 running
121. Process foo with pid 4 running
122. Process foo with pid 3 running
123. Process foo with pid 4 running
124. Process foo with pid 3 running
125. Process foo with pid 3 running
126. Process foo with pid 3 running
127. Process foo with pid 3 running
128. Process foo with pid 3 running
129. Process foo with pid 3 running
130. Process foo with pid 3 running
131. Process foo with pid 3 running
132. Process foo with pid 3 running
133. Process foo with pid 3 running
134. Process foo with pid 3 running
135. Process foo with pid 3 running
136. Process init with pid 1 running
137. zombie!
138. Process sh with pid 2 running

ADDING time stamp

1. [004532176@jb359-2 xv6]$ make qemu-nox
2. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
3. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
4. -c -o console.o console.c
5. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
6. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
7. -c -o exec.o exec.c
8. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
9. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
10. -c -o fs.o fs.c
11. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
12. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
13. -c -o ide.o ide.c
14. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
15. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
16. -c -o lapic.o lapic.c
17. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
18. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
19. -c -o main.o main.c
20. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
21. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
22. -c -o mp.o mp.c
23. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
24. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
25. -c -o pipe.o pipe.c
26. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
27. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
28. -c -o proc.o proc.c
29. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
30. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
31. -c -o sleeplock.o sleeplock.c
32. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
33. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
34. -c -o spinlock.o spinlock.c
35. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
36. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
37. -c -o syscall.o syscall.c
38. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
39. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
40. -c -o sysfile.o sysfile.c
41. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
42. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
43. -c -o sysproc.o sysproc.c
44. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
45. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
46. -c -o trap.o trap.c
47. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
48. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
49. -c -o uart.o uart.c
50. gcc -fno-pic -**static** -fno-builtin -fno-strict-aliasing -O2 -Wall
51. -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stackprotector
52. -c -o vm.o vm.c
53. ld -m elf\_i386 -T kernel.ld -o kernel entry.o bio.o console.o
54. exec.o file.o fs.o ide.o ioapic.o kalloc.o kbd.o lapic.o log.o
55. main.o mp.o picirq.o pipe.o proc.o sleeplock.o spinlock.o
56. string.o swtch.o syscall.o sysfile.o sysproc.o timer.o trapasm.o
57. trap.o uart.o vectors.o vm.o -b binary initcode entryother
58. objdump -S kernel > kernel.asm
59. objdump -t kernel | sed '1,/SYMBOL TABLE/d; s/ .\* / /; /^$/d' >
60. kernel.sym
61. dd **if**=/dev/zero of=xv6.img count=10000
62. 10000+0 records in
63. 10000+0 records out
64. 5120000 bytes (5.1 MB) copied, 0.116044 s, 44.1 MB/s
65. dd **if**=bootblock of=xv6.img conv=notrunc
66. 1+0 records in
67. 1+0 records out
68. 512 bytes (512 B) copied, 0.000914179 s, 560 kB/s
69. dd **if**=kernel of=xv6.img seek=1 conv=notrunc
70. 357+1 records in
71. 357+1 records out
72. 183072 bytes (183 kB) copied, 0.00410289 s, 44.6 MB/s
73. which: no qemu in
74. (/usr/local/bin:/opt/eclipse:/opt/scilab/bin:/opt/androidstudio/bin:/opt/argouml:/usr/lib64/openmpi/bin:/usr/local/cuda/b
75. in:/share/bin:/opt/Xilinx/14.7/ISE\_DS/ISE/bin/lin64:/opt/Xilinx/
76. 14.7/ISE\_DS/common/bin/lin64:/opt/android-sdklinux/tools:/opt/android-sdk-linux/platformtools:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/u/cse/0
77. 04867222/bin/.)
78. qemu-system-i386 -nographic -drive
79. file=fs.img,index=1,media=disk,format=raw -drive
80. file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
81. (process:28641): GLib-WARNING \*\*: gmem.c:482: custom memory
82. allocation vtable not supported
83. xv6...
84. cpu1: starting
85. cpu0: starting
86. Process initcode with pid 1 running with createTime 0
87. Process initcode with pid 1 running with createTime 0
88. Process initcode with pid 1 running with createTime 0
89. Process initcode with pid 1 running with createTime 0
90. Process initcode with pid 1 running with createTime 0
91. Process initcode with pid 1 running with createTime 0
92. Process initcode with pid 1 running with createTime 0
93. Process initcode with pid 1 running with createTime 0
94. Process initcode with pid 1 running with createTime 0
95. Process initcode with pid 1 running with createTime 0
96. Process initcode with pid 1 running with createTime 0
97. Process initcode with pid 1 running with createTime 0
98. Process initcode with pid 1 running with createTime 0
99. Process initcode with pid 1 running with createTime 0
100. Process initcode with pid 1 running with createTime 0
101. Process initcode with pid 1 running with createTime 0
102. Process initcode with pid 1 running with createTime 0
103. Process initcode with pid 1 running with createTime 0
104. Process initcode with pid 1 running with createTime 0
105. Process initcode with pid 1 running with createTime 0
106. Process initcode with pid 1 running with createTime 0
107. Process initcode with pid 1 running with createTime 0
108. Process initcode with pid 1 running with createTime 0
109. Process initcode with pid 1 running with createTime 0
110. Process initcode with pid 1 running with createTime 0
111. Process initcode with pid 1 running with createTime 0
112. Process initcode with pid 1 running with createTime 0
113. Process initcode with pid 1 running with createTime 0
114. Process initcode with pid 1 running with createTime 0
115. Process initcode with pid 1 running with createTime 0
116. Process initcode with pid 1 running with createTime 0
117. Process initcode with pid 1 running with createTime 0
118. Process initcode with pid 1 running with createTime 0
119. Process initcode with pid 1 running with createTime 0
120. Process initcode with pid 1 running with createTime 0
121. Process initcode with pid 1 running with createTime 0
122. Process initcode with pid 1 running with createTime 0
123. Process initcode with pid 1 running with createTime 0
124. Process initcode with pid 1 running with createTime 0
125. Process initcode with pid 1 running with createTime 0
126. Process initcode with pid 1 running with createTime 0
127. Process initcode with pid 1 running with createTime 0
128. Process initcode with pid 1 running with createTime 0
129. Process initcode with pid 1 running with createTime 0
130. Process initcode with pid 1 running with createTime 0
131. Process initcode with pid 1 running with createTime 0
132. Process initcode with pid 1 running with createTime 0
133. Process initcode with pid 1 running with createTime 0
134. Process initcode with pid 1 running with createTime 0
135. Process initcode with pid 1 running with createTime 0
136. Process initcode with pid 1 running with createTime 0
137. P

Evaluation: In this lab, I had done everything that was required. I cut some of the typescript down when running xv6 because it was super long. It would have been about 20+ pages if I kept it all and it just said the same thing, so I thought it was pointless to have. It was a difficult lab because I could not figure out semaphores and editing my files. The most difficult part was editing my makefile because I was not present in class about how we edited the makefile. At the end, I finished it. I would give myself a full score of 20/20.

Score: 20/20