Introduction to Operating Systems

Lab 2: Pintos & Project 1

GAO Ming

SE@ECNU
(for course related communications)
mgao@sei.ecnu.edu.cn

Mar. 19, 2014

Outline

Pintos

- Introduction/manual:
  `http://www.stanford.edu/class/cs140/projects/pintos/pintos_1.html`
- PDF manual:
  `http://www.stanford.edu/class/cs140/projects/pintos/pintos.pdf`
- All things you need:
  `http://www.stanford.edu/class/cs140/projects/pintos/`
- Source:
  `http://www.stanford.edu/class/cs140/projects/pintos/pintos.tar.gz`

Install Pintos in OSProj Virtual Machine

Pls. refer to materials in past courses.

- 2009_lab_1.pdf
- 2009_lab_2.pdf

Install Pintos

- `http://www.stanford.edu/class/cs140/projects/pintos/pintos_12.html`

Outline

Data structures

- Read thread related source code
  - threads/init.c
  - threads/thread.h, thread.c
  - threads/switch.h
  - threads/synch.h, synch.c
  - devices/timer.c, timer.h

Synchronization

Solution that disable interrupts:

- threads/synch.h, synch.c

Outline

Alarm clock

Reimplement: void timer_sleep (int64_t ticks)

- In devices/timer.c
- Without using busy waiting

Priority scheduling

- Implement priority scheduling
- Implement priority donation (for locks)
- Implement set/get priority functions

Introduction to the 4.4BSD scheduler

Multi-level feed-back queue scheduling

- priority = PRI_MAX - (recent_cpu / 4) - (nice * 2)
- recent_cpu = (2*load_avg)/(2*load_avg + 1) * recent_cpu + nice
- load_avg = (59/60)*load_avg + (1/60)*ready_threads

Notes

- No floating-point arithmetic in the kernel
- Assume that x and y are fixed-point numbers, and n is an integer. Fixed point numbers are in signed $p.q$ format, where $p + q = 31$, and f is $1 << q$:

convert n to fixed point	$n * f$
convert x to integer (rounding toward zero)	x/f
convert x to integer (rounding toward nearest)	$(x + f/2)/f$ if $x >= 0$, $(x - f/2)/f$ if $x <= 0$
add x and y	$x + y$
substract y from x	$x - y$
add x and n	$x + n * f$
substract n from x	$x - n * f$
multiply x by y	$((int64\_t)x) * y/f$
multiply x by n	$x * y$
divide x by y	$((int64\_t)x) * f/y$
divide x by n	x/y

Outline

Lab report

1. Data structures: see Manual Appendix D
2. Algorithms: see Manual Appendix D
3. Synchronization: see Manual Appendix D
4. Rationale: see Manual Appendix D
5. Known errors: the summary of the testing, and your explanation on failed tests

Hand-ins and deadline

- All hand-ins (code and lab report) should be received before/on April 13, 2014 (before the end of our class)
- What to submit?
    - A lab report: by hand with a printed attachment on A4 papers
    - An zipped package (the file name is xxxxxxxx.zip, where xxxxxxxx is your full student id) with following files should be sent to my gmail address (os.sei.ecnu@gmail.com) vis an email with title: proj1_submit:
        - The lab report attachment: in plain txt format, in English, in the root directory of the zipped file
        - All source code files you modified or added: in relative path corresponding to pintos/src
          e.g. if you've modified synch.c in pintos/src/threads, then the file synch.c should appear in /threads of the package
        - A readme file states all things that I should notice on your submission. It could be left as a blank file if you have nothing to say. But the file **must** exist.

Scoring

$\frac{P}{P+F} \times 50\% + S \times 50\%$

- P: number of items passed the test
- F: number of items failed in the test
- S: score on your lab report

Note:

- Inconsistency between your implementation and your report will increase F and decrease P.
- Copy other's code is not allowed.
- Cheating is not allowed.

Outline

Ctags

```
>cd  /Desktop/pintos/src
>ctags -R *
```

- Add the following two lines to /.vimrc:

  ```
  set nu
  set tags= /Desktop/pintos/src/tags
  ```

Testing

```
>cd threads
>make check
```

gdb

```
>cd threads/build
>pintos –gdb – run multi-alarm
```

- Open another terminal

```
>pintos-gdb kernel.o
(gdb) target remote localhost:1234
```

- Then, you may be able to debug pintos in gdb
    - You may omit the following warning:

      ```
      warning: Remote failure reply: Eff
      0x0000fff0 in ?? ()
      ```

Pintos manual Chapter 2 and Appendix A, B, D, and E.