VATSAL SONI - 2018201005
DARSHAN KANSAGARA -2018201033

Pintos Report

Operating System (PG), Monsoon-2018

1. Installation

a. Install Qemu simulator to run pintos.
   i. *sudo apt-get install qemu.*
b. Download Pintos.
c. Change GDBMACROS.
   i. Change the file at location /$HOME/os-pintos/pintos/src/utils/pintos-gdb.
   ii. GDBMACROS=$HOME/os-pintos/pintos/src/misc/gdb-macros.
d. Edit Makefile in utils directory.
   i. Replace line number five, “LDFLAGS = -lm” by “LDLIBS = -lm”.
   ii. Compile the utilities in the “utils” folder using “make” command.
e. Edit Make.vars in threads directory.
   i. Change the last line to “SIMULATOR = –qemu”.
f. Compile the Pintos kernel using “make” command.
g. Edit pintos file.
   i. Change line no. 103 to “$sim = “qemu” if !defined $sim;” to enforce qemu as simulator.
   ii. On line no. 259, replace “kernel.bin” to path pointing to kernel.bin file at “$HOME/os-pintos/pintos/src/threads/build/kernel.bin”.
   iii. Change line no. 623 to ”my (@cmd) = (‘qemu-system-x86_64’);”
h. Edit Pintos.pm file.
   i. Replace “loader.bin” at line no.362 to the path “$HOME/os-pintos/pintos/src/threads/build/loader.bin”.
i. Run Pintos.
   i. *pintos run alarm-multiple.*
j. Add to PATH variable to run pintos from anywhere.
   i. Add “export PATH=$HOME/os-pintos/pintos/src/utils:$PATH” in this file HOME/.bashrc

2. Debugging tool installation

a. To install cscope
   i. sudo apt-get install cscope ctags vim
b. To run
   i. $cscope –Rvkq

3. Hello world test

a. Create hellotest file in pintos/src/tests/threads/hellotest.c.
b. Define helloworld function in test.h file.
c. Implement helloworld function in hellotest file.
d. Add helloworld function and file name in test.c file.
e. Run “pintos run hellotest” command.
f. **Modified Files** : test.h, test.c

4. Pre-emption of threads

a. Solution: Add thread to sleep list for given sleep time and change its status to THREAD_BLOCK. Wake up thread when sleep time expires and add it in ready queue and change its status to THREAD_READY. At every second thread_tick() function is called by timer interrupt function which calls thread_wake() function. Thread_wake() function traverse through whole sleep list and check weather thread’s sleep time expired or not. If it has been expired then remove it from sleep list and add it into the ready list.

b. Modifiled files :

i. timer.h, timer.c, thread.h, thread.c

5. Implementing Priority Scheduling

a. Solution: Maintain ready list in sorted order based on priority in descending order. So every time after inserting thread in ready list, sort ready list. In thread_set_priority() function whenever priority of running thread is changed then schedule the thread which has higher priority among first thread(from ready list which has highest priority) and currently running thread. Also maintain waiting list of threads in sorted order (which are waiting for semaphore) based on priority.

b. Modified files:

i. thread.h, thread.c, sync.h, sync.c

6. References

a. https://web.stanford.edu/class/cs140/projects/pintos/pintos_6.html
b. https://tssurya.wordpress.com/2014/08/16/installing-pintos-on-your-machine/
c. https://pintosiiith.wordpress.com/2012/10/01/running-test-cases-for-pintos-assignment/