Lab 2 - Recitation

18-648

Periodic App

- Busy loop for C out of T in every T period
- Has to be suspended after executing for C time units
  - Suspending using sleep loop is not valid ;)
- Make sure autohotplug is turned off
  - Make sure `sched_setaffinity` works in both kernel and userspace
- Virtual vs real timers when working with itimers in userspace
  - Difference between **`ITIMER_REAL`**, **`ITIMER_VIRTUAL`** and **`ITIMER_PROF`**

Scheduler

- Each core has its own runqueue
- On 1 core, a single instruction stream is executing at a time
- `__schedule()` function saves some state and switches from the current to the next task in the runqueue
- Any data structures you manipulate in this function, must be properly protected
  - Remember you are on a multi-core platform!
- Make sure any variables in your `task_struct` are initialized appropriately
  - Children of a thread in the reservation framework should not start off in the reservation framework
- Look up how the linux kernel implements priorities

Signals

- Useful doc: http://man7.org/linux/man-pages/man7/signal.7.html
- `SIGEXCESS` is a new signal that you are sending from the kernel
  - If we write a new application, the kernel headers you export should allow our app to see a signal named “`SIGEXCESS`”.
  - Exporting header from kernel to user-space takes place with make headers_install
  - If you had issues with your calc application not building during demo because you changed kernel/usr/include note this
  - Remember, the include file(s) that you edit will be in an **architecture specific** location

Locks

- Difference between mutex & spinlock
- `spin_lock_irq` vs `spin_lock_irqsave`: https://stackoverflow.com/questions/2559602/spin-lock-irqsave-vs-spin-lock-irq
- `rcu_read_lock`: https://www.kernel.org/doc/Documentation/RCU/whatisRCU.txt
- Be careful when allocating memory while holding locks. Hint: Think about `kmalloc`’s parameters

Makefile

- Follow documentation at the end of the handout for Lab 2
- Assume we just have Lab 0 setup, and none of your custom env variables
- Header paths inside your app Makefile should be relative to the app directory's position in the kernel

General Tips

- Be careful when performing 64-bit division in the kernel. You might get an error or a warning about this. Hint: https://lists.kernelnewbies.org/pipermail/kernelnewbies/2011-May/001870.html
- Useful Resources
    - https://lwn.net
    - http://elixir.free-electrons.com/linux/v3.1.10/source
    - https://www.amazon.com/Linux-Kernel-Development-Robert-Love/dp/0672329468