Teaching Using Multi-Core Xinu

Priya Bansal and Dennis Brylow

XinuPi3?

XinuPi3 is the first open source lightweight instructional operating system that uses multiple cores

- **It is much easier and much more effective to teach using a <u>real</u> operating system rather than a simulation**
- **It is more important than ever to teach multi-core concepts to the younger generation,**
  - **Even smartphones run on multiple cores**
- **XinuPi3 can assist, not only in teaching these very important multi-core concepts, but also in teaching typical operating system concepts in a new way.**

How it looks

Running a core is as simple as:

```
*(volatile fn *)(CORE_MBOX_BASE +
CORE_MBOX_OFFSET * num) = Core1Setup;
```

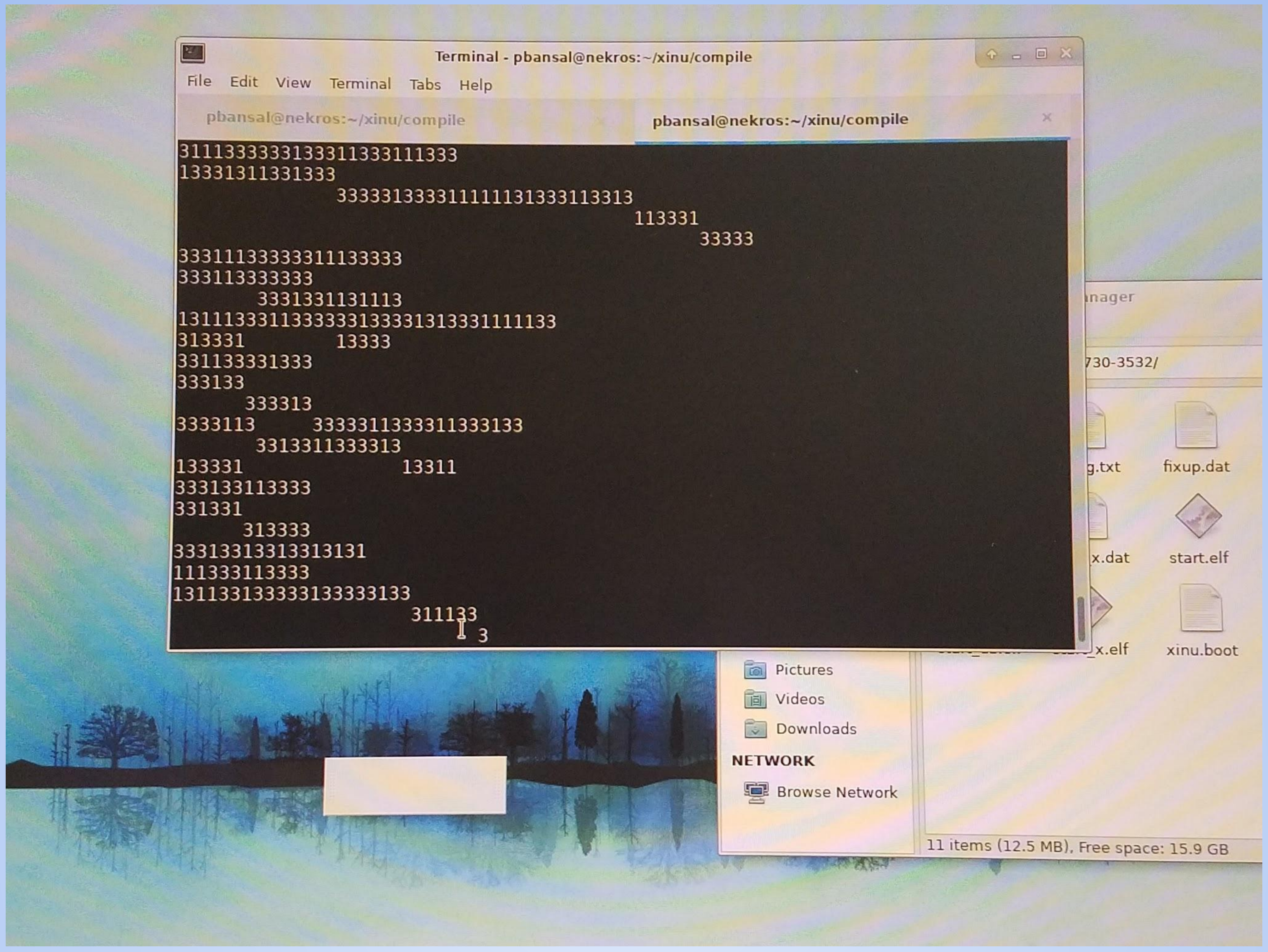
This line of code instructs a core numbered 1 - 3 to execute at address Core1Setup

Once a core has been awoken,

```
ldr r0, =core_init_sp
ldr sp, [r0, #4]
```

The stack must be set up in ARM assembly code. This is the only requirement needed to start running code on a core, but if this requirement isn't *immediately* met, the core will be stuck.

Using just 3 lines of code (in reality, only 2 of those three lines are needed), any student can run code on multiple cores. Cool!


Terminal - pbansal@nekros:~/xinu/compile
File Edit View Terminal Tabs Help
pbansal@nekros:~/xinu/compile
pbansal@nekros:~/xinu/compile

```
31113333331333113331113 33
13331311331333
              3333313333111111313331133 13
                                          113331
                                                33333
3331113333331113 3333
333113333333
       3331331131113
13111333113333331333313133 31111133
313331        13333
331133331333
333133
      333313
3333113     33333113333113 33133
       3313311333313
133331            13311
333133113333
331331
      313333
33313313313313131
111333113333
13113313333313333313 3
                    311133
                       3
```

Numerical soup, caused by multiple cores printing at the same time

It's Easier

- **Can teach concepts about processes without having to first teach how to switch between processes**
- **Easier to think about processes running concurrently**
  - **Helps students understand why mutexes are necessary and how they work**

It's Harder

- **Mutexes and semaphores don't work the same way**
- **Harder to control what is running and when**
- **Processes *actually* running at the same time, which is cool but can get messy**

It's Worth It

- **Old ways of using mutexes and semaphores will become unnecessary to know**
- **Students become more satisfied if knowing they are working on new, unchartered territory**
- **Employers would rather hire employers with experience with multi-core systems**

References

Brown, Richard, Elizabeth Shoop, Joel Adams, Curtis Clifton, Mark Gardner, Michael Haupt, and Peter Hinsbeeck. "Strategies for Preparing Computer Science Students for the Multicore World." Proceedings of the 2010 ITiCSE Working Group Reports on Working Group Reports - ITiCSE-WGR '10 (2010): n. pag. Web. 26 July 2017.

Yazici, Ali, Alok Mishra, and Ziya Karakaya. "Teaching Parallel Computing Concepts Using Real-Life Applications." International Journal of Engineering Education 32.2 (2016): 772-81. ResearchGate. Web. 26 July 2017.

Acknowledgments

Special Thanks To:

- Phil Elwell for helping us get started
- David Welch for teaching about multi-cores
- Goswin von Brederlow for much-needed code