Preemptive Multicore Scheduling on Embedded Xinu

Brian Weithers, Dr. Dennis Brylow

Introduction

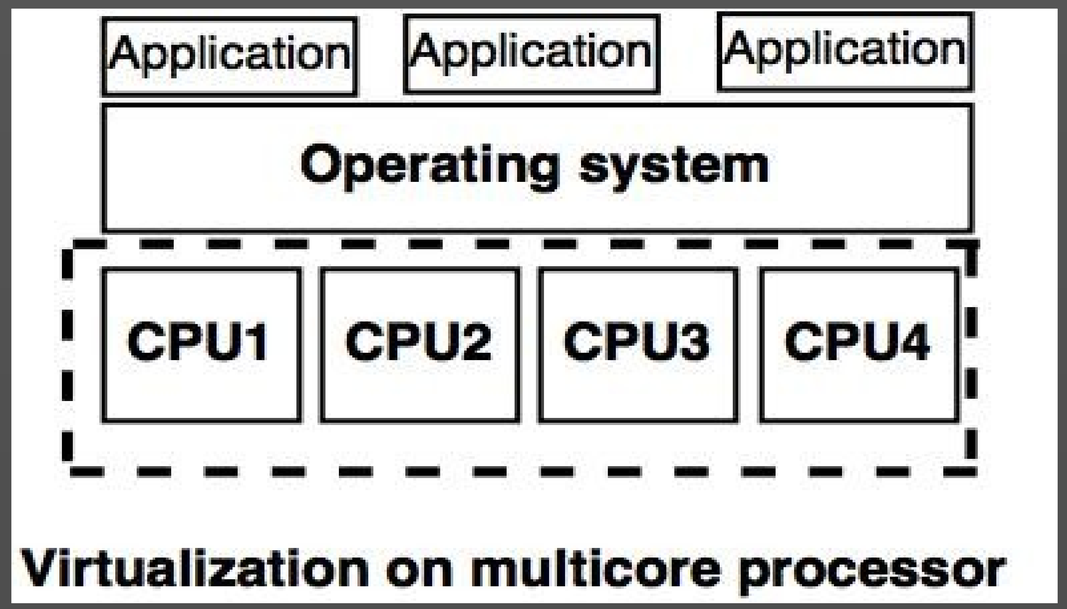
- Need education in OS concepts, but commercial OS's are too complex for intro level
- Create simple OS for education: Xinu
- Hardware constantly changing
- Creates a constant need for updated teaching materials

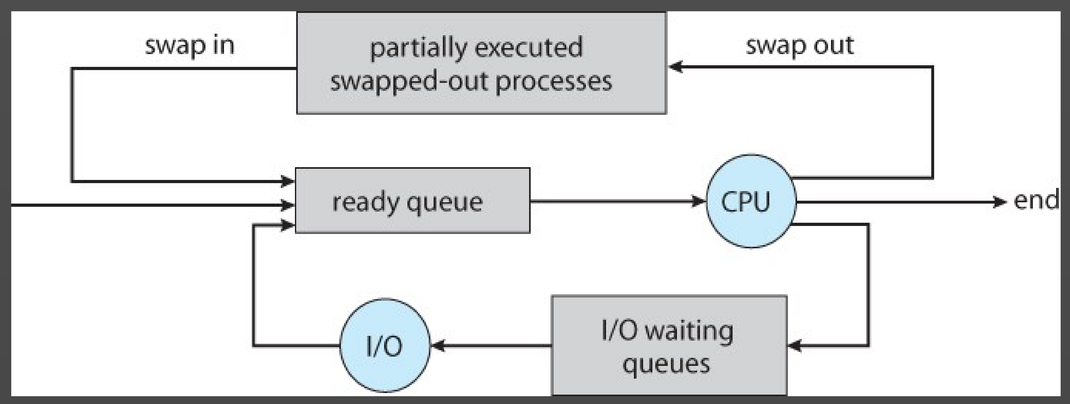
Objective

- Update a part of Xinu to match new multicore hardware: The process scheduler
- Use updated Xinu to teach new OS concepts

Acknowledgements:

The National Science Foundation, Benjamin Levandowski, Rade Latinovich






```c
/*
 * Defines what an entry in thread table looks like
 */
struct thread
{
    int core;           // core thread last ran on
    int state;          // state of thread e.g. THRCURR
    int prio;           // thread priority
    char name[LENGTH];  // name of thread
    thr_id parent;      // ID num of parent thread
    void *stkptr;       // thread stack pointer
};
```

Results

- Working simple, preemptive scheduler
- Change OS Assignment 4 to include multicore scheduling instead of single core
- Mask most difficult details, to not increase complexity for students
- Introduce and familiarize students with modern multicore concepts

Future Work

- Design Operating Systems curriculum surrounding multicore scheduling
- Improve scheduler algorithm