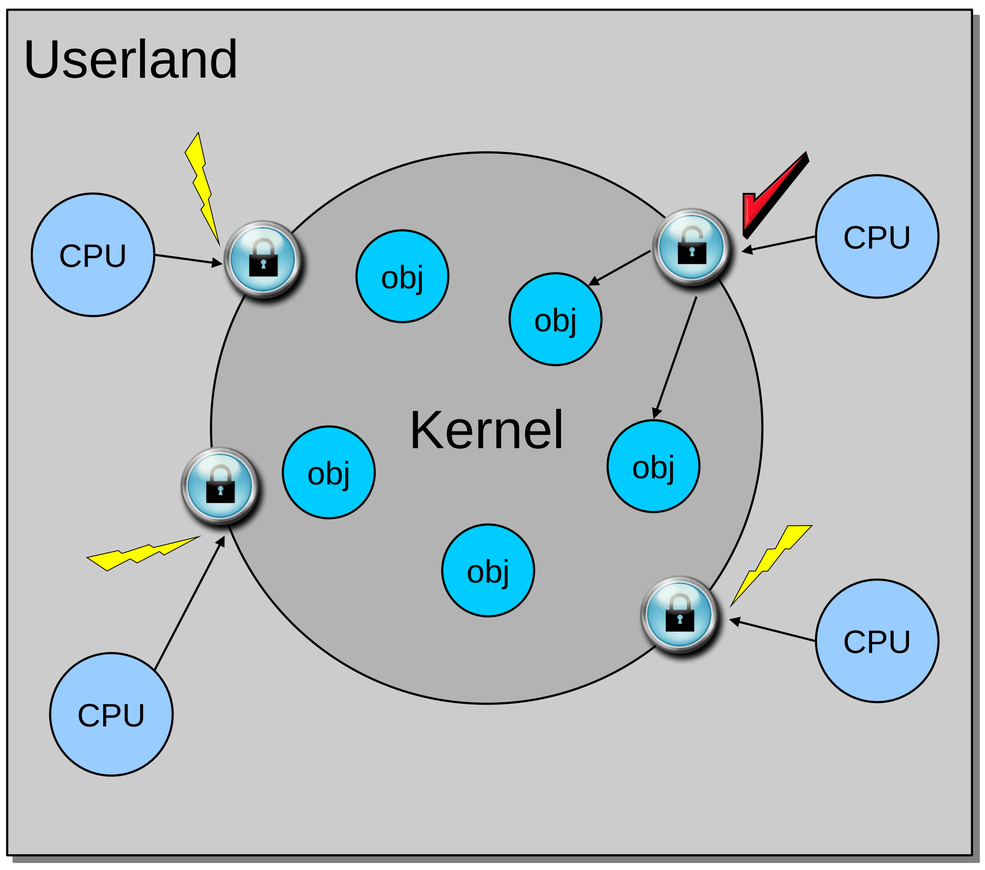
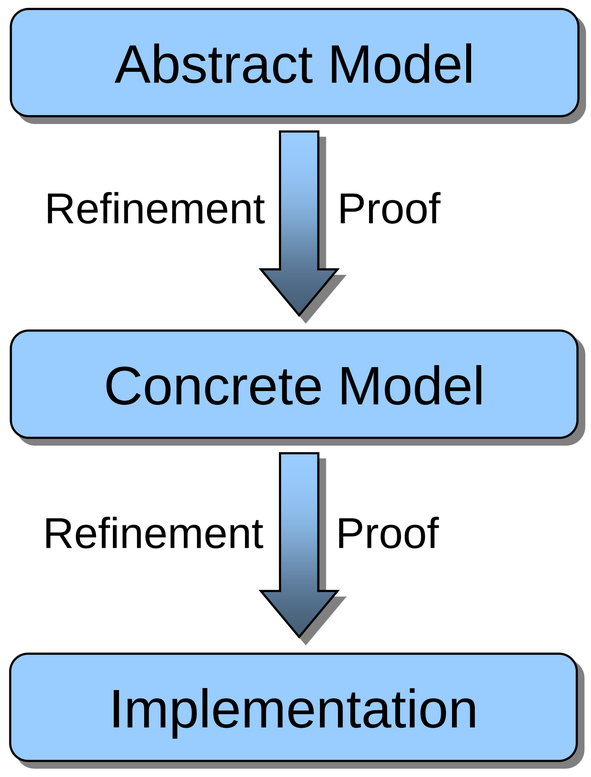
Towards a Formally Verifiable Multiprocessor Microkernel

Michael von Tessin (NICTA and UNSW)

NICTA

L4.verified






```
schedule ≡ do                                   Isabelle/HOL
  Threads ← allActiveTCBs;
  thread ← select threads;
  switch_to_thread thread
od
```

```
schedule = do                                   Haskell
  curThread <- getCurThread
  runnable <- isRunnable curThread
  Time <- threadGet tcbTimeSlice curThread
  when (not runnable || time == 0)
    chooseThread
```

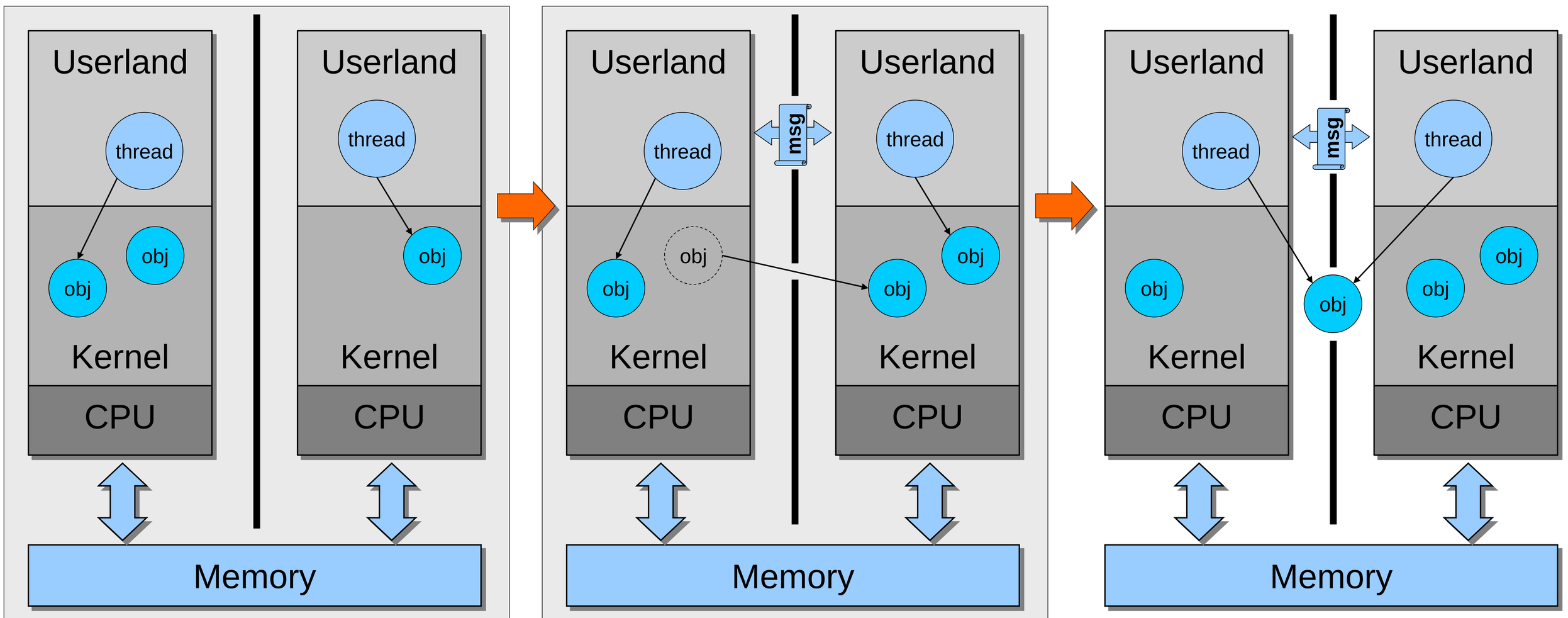
```
void schedule()                                 C
{
  if (!isRunnable(ksCurThread) ||
        ksCurThread->tcbTimeSlice == 0)
    chooseThread();
}
```

Reduce parallelism by having a **big lock** around the whole kernel

Formal verification approach of the **uniprocessor** version

Reduce concurrency → **Reduce proof complexity induced by concurrency** → + → **GOAL!** ✓

Reduce sharing with a **multikernel** approach like in *Corey* or *Barrelfish*



Progress: implemented in C and semi-formally proved

low-level design