8.

```
struct cpu*
mycpu(void)
{
  int apicid, i;
  if(readeflags() & FL_IF)
    panic("mycpu called with interrupts
    enabled\n");
  apicid = lapicid();
  for(i=0; i<ncpu; ++i)
  {
    if(cpus[i].apicid == apicid)
      return &cpus[i];
  }
  panic("unknown apicid\n");
}
struct proc *
myproc(void)
{
  struct cpu *c;
  struct proc *p;
  pushcli();
  c = mycpu();
  p = c->proc;
  popcli();
  return p;
}
```

In many places, the processor must identify which process it is running, for example when a processor serves a yield system call, the processor must determine which process is invoking yield. xv6 follows the standard plan that relies on a tiny

bit of hardware support.

The kernel uses mycpu to find its apicid, which in turn calls the function lapicid, The function lapicid returns the hardware identifier for this processor. mycpu uses identifier to find its struct cpu.

The kernel uses myproc to find struct proc for the process that is running on current processor. The function myproc disables interrupt and invokes mycpu to find its processor state, which contains a field proc.