**Which is the scheduling algorithm for XV6? Support with code.**

This scheduling algorithm is Round-Robin, because at first if a hardware interrupt is waiting to be handled, the scheduler’s CPU will handle it before continuing the process. Then it checks in the process table for a runnable process and after getting one we set up the descriptors and the current process task state. After doing this it marks the process as it were running and it calls switch() to start running this new process.

void scheduler(void)

{

struct proc \*p;

for(;;){

// Enable interrupts on this processor.

sti();

// Loop over process table looking for process to run.

acquire(&ptable.lock);

for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){

if(p->state != RUNNABLE)

continue;

// Switch to chosen process. It is the process's job

// to release ptable.lock and then reacquire it

// before jumping back to us.

proc = p;

switchuvm(p);

p->state = RUNNING;

swtch(&cpu->scheduler, proc->context);

switchkvm();

// Process is done running for now.

// It should have changed its p->state before coming back.

proc = 0;

}

release(&ptable.lock);

}

}



