

Drugi kolokvijum iz Operativnih sistema 1

avgust 2026.

1. (10 poena)

```
void Semaphore::wait () {
    if (this->val-- <= 0) {
        this->waiting.add(ContextSwitch::instance()->getRunning());
        ContextSwitch::instance()->suspendRunning();
    }
}

void Semaphore::signal () {
    if (this->val++ < 0) {
        Task* resumed = this->waiting.get();
        ContextSwitch::instance()->preemptRunning(resumed);
    }
}

void Timing::wakeup () {
    while (this->now >= this->sleeping.getFirst()->getWakeupTime()) {
        Task* resumed = this->sleeping.removeFirst();
        ContextSwitch::instance()->preemptRunning(resumed);
    }
}
```

2. (10 poena)

```
#include <fcntl.h>
#include <sys/stat.h>
#include <semaphore.h>

class Mutex {
public:
    Mutex (const char *name);
    ~Mutex ();
    int entry ();
    int exit();
private:
    sem_t* mutex = 0;
    pid_t pid = 0;
};

Mutex::Mutex (const char *name) {
    this->mutex = sem_open(name,O_CREAT,1);
}

int Mutex::entry () {
    pid_t p = getpid();
    if (!this->mutex || this->pid==p) return -1;
    int ret = sem_wait(this->mutex);
    if (ret<0) return ret;
    this->pid = p; // Must not be done before sem_wait
    return 0;
}

int Mutex::exit () {
    if (!this->mutex || this->pid!=getpid()) return -1; // error
    this->pid = 0;
    return sem_post(this->mutex);
}

Mutex::~~Mutex () {
    sem_close(this->mutex);
}
```