

# Rešenja prvog kolokvijuma iz Operativnih sistema 2 novembar 2022.

## 1. (10 poena)

U strukturu PCB dodaje se polje entryTS tipa Time koje čuva trenutak ulaska procesa u red spremnih. Izmene klase Scheduler su samo u operacijama get i put i napisane su podvučeno:

```
PCB* Scheduler::put (PCB* p, bool wasBlocked, Time execTime) {
    if (pcb==0) return; // Exception!
    if (wasBlocked)
        pcb->execTime = 0;
    else
        pcb->execTime += execTime;
    unsigned long pri = pcb->execTime/TIME_PRI_RESOL;
    if (pri>MaxPri) pri = MaxPri;
    ready[pri].put(p);
    count++;
    pcb->entryTS = Timing::getTime(); // Timestamp of the entry
}

PCB* Scheduler::get () {
    for (int i=0; i<=MaxPri; i++) {
        PCB* p = ready[i].get();
        if (p) {
            Time ts = Timing::getTime(); // Timestamp of the exit
            pcb->timeSlice = (ts-pcb->entryTS)/count;
            count--;
            return p;
        }
    }
    return 0;
}
```

## 2. (10 poena)

```
inline void Condition::wait () {
    waitingCond.put(Thread::running);
    mc->open();
    Thread::dispatch();
}

inline void Condition::signal () {
    Thread* t = waitingCond.get();
    if (!t) return;
    Scheduler::put(t);
    mc->waitingToContinue = Thread::running;
    Thread::dispatch();
}
```

### 3. (10 poena)

```
public class Client {
    private static String[] SERVERS = {"srv1.etf.bg.rs", "srv2.etf.bg.rs"};
    private int nextServer = 0;

    public void processDataFromServer() {
        new ClientHandler(this).start();
    }

    public void processData(String data) {
        // Data processing
    }

    public Service getService() throws IOException {
        int server;
        synchronized (this) {
            server = nextServer;
            nextServer = (nextServer == SERVERS.length) ? 0 : nextServer +
1;
        }

        Socket socket = new Socket(SERVERS[server], 5555);
        return new Service(socket);
    }
}

public class ClientHandler extends Thread {
    private Client client;

    public ClientHandler(Client client) {
        this.client = client;
    }

    public void run() {
        try {
            Service service = client.getService();

            service.sendMessage("GET_DATA");
            String result = "";
            for (String msg = service.receiveMessage(); msg != null; msg =
service.receiveMessage()) {
                result += msg;
            }

            client.processData(result);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Klasa Service je data na vežbama.