

# Rešenja prvog kolokvijuma iz Operativnih sistema 2 februar 2026.

## 1. (10 poena)

```
void Timing::setIntHndlProc () {
    int minproc = 0;
    Priority minpri = Running::get(minproc)->getPri();
    for (int p=1; p<NUM_OF_PROCESSORS; p++) {
        Priority pri = Running::get(p)->getPri();
        if (pri<minpri)
            minpri = pri, minproc = p;
    }
    IntCntl::setIntHndlProc(minproc);
}
```

2. (10 poena) Ukoliko monitor, nakon što operacija *signal* deblokira proces koji je na uslovnoj promenljivoj čekao, prioritira izvršavanje tog deblokiranog procesa u odnosu na onog koji je uradio *signal*, postoji sledeći problem utrivanja (*race condition*). Ako je bafer bio prazan, nakon što proizvođač uradi `signal(itemAvailable)`, aktivira se potrošač koji je čekao na ovoj uslovnoj promenljivoj. Potrošač preuzme postavljeni zahtev i uradi `signal(taken[base])` koji nema efekta, jer se na ovoj uslovnoj promenljivoj proizvođač još nije suspendovao. Potom se proizvođač nepotrebno suspenduje na operaciji `wait(taken[oldTop])`.

Dodatni problem je što se `base` menja pre poziva operacije `signal`, pa se ne signalizira odgovarajuća uslovna promenljiva.

## 3. (10 poena)

Klijent na kraju treba da pošalje poruku da nema više linija:

```
service.sendMessage("UPLOAD#0");
```

Klijent ne treba da ponavlja slanje već poslatih linija, pa sledeća linija treba da stoji unutar petlje:

```
List<String> lines = new ArrayList<>();
```

Server na početku treba da preuzme ime fajla i to ime koristi za otvaranje fajla:

```
String fileNameMsg = service.receiveMsg();
File file = new File(fileNameMsg);
```

Server treba da provera da li je završen transfer fajla:

```
if (lineCount == 0) {
    break;
}
```