Drugi kolokvijum iz Operativnih sistema 1
Avgust 2023.

1. (10 poena)

a)(7)

```
int max (Node* nd) {
  int m = nd->val, m1;
  if (nd->right)
    if (fork()==0)
      exit(max(nd->right));
  if (nd->left) {
    m1 = max(nd->left);
    if (m1>m) m = m1;
  }
  if (nd->right) {
    wait(&m1);
    if (m1>m) m = m1;
  }
  return m;
};
```

b)(3) Prvi nivo (koren stabla) obrađuje jedan, inicijalni proces. Drugi nivo, osim njega, obrađuje još jedan proces, njegovo dete. Svaki nivo obrađuju svi procesi koji obrađuju prethodni nivo i još toliko novih procesa njihove dece, odnosno dvostruko više (nijedan se ne gasi). Tako n-ti nivo obrađuju svi kreirani procesi, njih 2^{n-1}.

2. (10 poena)

a)(7) U klasu `Thread` dodaje se privatni podatak član `isSuspended` tipa `bool` inicijalizovan na `false`.

```
void Thread::suspend () {
  lock();
  runningThread->isSuspended = true;
  if (setjmp(runningThread->context)==0) {
    runningThread = Scheduler::get();
    longjmp(runningThread->context,1);
  }
  unlock();
}

void Thread::resume () {
  lock();
  if (this->isSuspended) {
    this->isSuspended = false;
    Scheduler::put(this);
  }
  unlock();
}
```

b)(3) Opisano rešenje ima problem utrkivanja (*race condition*) koje se može dogoditi na sledeći način: uslov nije ispunjen; nit koja treba da čeka na uslov ispituje uslov i zaključuje da treba da se suspenduje, ali pre nego što to uradi nit koja signalizira uslov uradi *resume* bez efekta, pa se nakon toga prva nit bespotrebno suspenduje potencijalno neograničeno.

3. (10 poena)

```
class OptimisticCCtrl {
public:
```

```
  OptimisticCCtrl () {}

  Data* startTrans (Data** original);
  bool  commit ();
private:
  Data **shared = 0, *read = 0, *copy = 0;
};

Data* OptimisticCCtrl::startTrans (Data** original) {
  shared = original;
  read = *original;
  return copy = new Data(*read);
}

bool OptimisticCCtrl::commit () {
  bool ret = cmp_and_swap(shared,read,copy);
  if (!ret) delete copy;
  return ret;
}
```