

Rešenja zadatka za kolokvijum iz Operativnih sistema 1 septembar 2020.

1. (10 poena)

```
Thread* t_fork () {
    lock();

    // Allocate a new stack:
    void* stck = malloc(STACK_SIZE);
    if (!stck) throw ThreadCreationException();
    // and copy its contents from the parent's stack:
    memcpy(stck, Thread::running->stack, STACK_SIZE);

    // Create a new Thread object:
    Thread* newThr = new Thread();
    if (!newThr) { free(stck); throw ThreadCreationException(); }
    newThr->stack = stck;

    if (setjmp(newThr->context)==0) {
        // Parent thread:
        // and set its stack pointer:
        newThr->context->sp = newThr->context->sp -
            Thread::running->stack + stck;

        // Put the new thread to the ready list and return:
        Scheduler::put(newThr);
        unlock();
        return newThr;
    } else {
        // Child thread:
        unlock();
        return 0;
    }
}
```

2. (10 poena)

```
typedef void (*PF)(void*);

void cobegin (PF f[], void* af[], int n) {
    int* ids = new int[n];
    for (int i=0; i<n; i++)
        if ((ids[i]=fork())==0) {
            pf[i](af[i]);
            exit();
        }
    for (int i=0; i<n; i++)
        wait(ids[i]);
    delete ids;
}
```

3. (10 poena)

```
SegDesc* findSegDesc (SegDesc* root, size_t size) {
    SegDesc* sd = root;
    while (!sd) {
        if (sd->sz<=size) return sd;
        else sd = sd->right;
    }
    return nullptr;
}
```

4. (10 poena)

```
void truncateFile (FCB* fcb) {
    for (size_t i=0; i<SingleIndexSize; i++)
        if (fcb->singleIndex[i]) {
            freeBlock(fcb->singleIndex[i]);
            fcb->singleIndex[i] = 0;
        }

    if (fcb->dblIndex) {
        PBlock* dblIx = (PBlock*)getBlock(fcb->dblIndex);
        for (size_t i=0; i<DblIndexSize; i++)
            if (dblIx[i]) freeBlock(dblIx[i]);
        freeBlock(fcb->dblIndex);
        fcb->dblIndex = 0;
    }

    fcb->size = 0;
}
```