Rešenja zadataka za
kolokvijum iz Operativnih sistema 1
oktobar 2020.

### 1.	(10 poena)

```
#include <process.h>

int multispawn (int number, const char* path, const char* args[]){
  int ret = 0;
  // Prepare the arguments for the children:
  const char* childArgs[3];
  childArgs[0] = path;
  childArgs[2] = NULL;
  for (int i=0; i<number; i++) {
    childArgs[1] = args[i];
    // Create a child:
    int status = spawnvp(P_NOWAIT,path,childArgs);
    if (status>=0) ret++;
  }
  return ret;
}
```

### 2.	(10 poena)

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

### 3.	(10 poena)

```
extern IORequest* ioHead;
IORequest* pending = 0;

int startDMA () {
  pending = ioHead;
  if (pending==0) return 0;
  ioHead = ioHead->next;
  *dmaAddress = pending->buffer;
  *dmaCount = pending->size;
  return 1;
}

void transfer() {
  startDMA();
  *dmaCtrl = 1; // Start I/O
}
```

```
interrupt void dmaInterrupt () {
  if (pending==0) return; // Exception
  if (*dmaStatus&2) // Error in I/O
    pending->status = -1;
  else
    pending->status = 0;
  if (startDMA()==0) { // No more requests
    *dmaCtrl = 0;
    return;
  }
}
```

4. (10 poena)

```
size_t getPBlock (FCB* fcb, size_t bt) {
  if (bt>=fcb->size) return 0;
  size_t lBlk = bt/BLOCK_SIZE;
  size_t cluster = lBlk/CLUSTER_SIZE;
  size_t pBlk = fcb->index[cluster] + lBlk%CLUSTER_SIZE;
  return pBlk;
}
```