Rešenja zadataka za treći kolokvijum iz Operativnih sistema 1
Septembar 2014.

1. (10 poena)

```
char getchar () {
 static char buffer[BlockSize];
 static int cursor = BlockSize;
 if (cursor==BlockSize) {
   readBlock(buffer);
   cursor = 0;
 }
 return buffer[cursor++];
}
```

2. (10 poena)

```
class File {
public:
  File (const char *pathname, int flags, mode_t mode) throw Exception;
 ~File () throw Exception;

  void read  (byte* buffer, unsigned long size) throw Exception;
  void write (byte* buffer, unsigned long size) throw Exception;

private:
  int fh;
};

File::File (const char *p, int f, mode_t m) throw Exception : fh(0) {
  int s = open(p,f,m);
  if (s<0) throw Exception(s);
  fh=s;
}

File::~File () throw Exception {
  int s = close(fh);
  if (s<0) throw Exception(s);
}

void File::read (byte* b, unsigned long sz) throw Exception {
  int s = read(fh,b,sz);
  if (s<0) throw Exception(s);
}

void File::write (byte* b, unsigned long sz) throw Exception {
  int s = write(fh,b,sz);
  if (s<0) throw Exception(s);
}
```

3. (10 poena)

```
PBlock getFilePBlockNo (FCB* fcb, unsigned long bt) {
  if (fcb==0) return -1; // Exception
  unsigned long lblk = bt/BlockSize;  // Logical block number
  if (lblk<SingleIndexSize) return fcb->singleIndex[lblk];
  lblk -= SingleIndexSize;
  unsigned long dblIndex0Entry = lblk/DblIndex1Size;
  unsigned long dblIndex1Entry = lblk%DblIndex1Size;
  if (dblIndex0Entry>=DblIndex0Size)
    return -1; // Exception: file size overflow
  PBlock dblIndex1PBlkNo = fcb->dblIndex[dblIndex0Entry];
  PBlock* dblIndex1 = (PBlock*)getDiskBlock(dblIndex1PBlkNo);
  if (dblIndex1==0) return -1; // Exception
  return dblIndex1[dblIndex1Entry];
}
```