Rešenja drugog kolokvijuma iz Operativnih sistema 2, decembar 2016.

1. (10 poena) Sprečavanje mrtve blokade uvođenjem relacije totalnog uređenja resursa i alokacije resursa po monotono rastućem redosledu. Iz datog koda se može zaključiti sledeće: da bi mrtva blokada bila sprečena, mora biti: C>B, D; A>C, D; B, C>D. Odatle sledi: A>C>B>D. Tako:

```
Process X:              Process Y:              Process Z:
      req(D); req(B);         req(D); req(C);         req(D);
      req(C);                 req(A);                 req(B); req(C);
      rel(C);                 rel(A);                 rel(C); rel(B);
      rel(B); rel(D);         rel(C); rel(D);         rel(D);
```

2. (10 poena)

```
int getFreeFrame (uint pid, uint page, uint* frame) {
  int cur = 0;
  for (cur=0; cur<FreeFramePoolSize; cur++) {
    if (!freeFramePool[cur].isFree &&
          freeFramePool[cur].pid==pid && freeFramePool[cur].page==page) {
       // The frame of the same page found, return it and declare as free:
       *frame = freeFramePool[cur].frame;
       freeFramePool[cur].isSlotFree = 1;
       return 1;
    }
  }
  for (cur=0; cur<FreeFramePoolSize; cur++) {
    if (!freeFramePool[cur].isFree) {
       // Another free frame found, return it and declare as free:
       *frame = freeFramePool[cur].frame;
       freeFramePool[cur].isSlotFree = 1;
       return 0;
    }
  }
  // No free frame in the pool!
  return -1;
}
```

3. (10 poena)

```
ulong getWorkingSetSize (PCB* pcb) {
  if (pcb==0) return 0; // Exception
  ulong size = 0;
  for (int i=0; i<RefBitTableSize0; i++) {
    RefBitsTable1* refBitTablePtr = pcb->refBits[i];
    if (refBitTablePtr==0) continue;
    for (int j=0; j<RefBitTableSize1; j++) {
       RefBitReg ref = (*refBitTablePtr)[j];
       ref &= ~((~(RefBitReg)0)>>NumOfHistoryBits);
       if (ref) size++;
    }
  }
  return size;
}
```