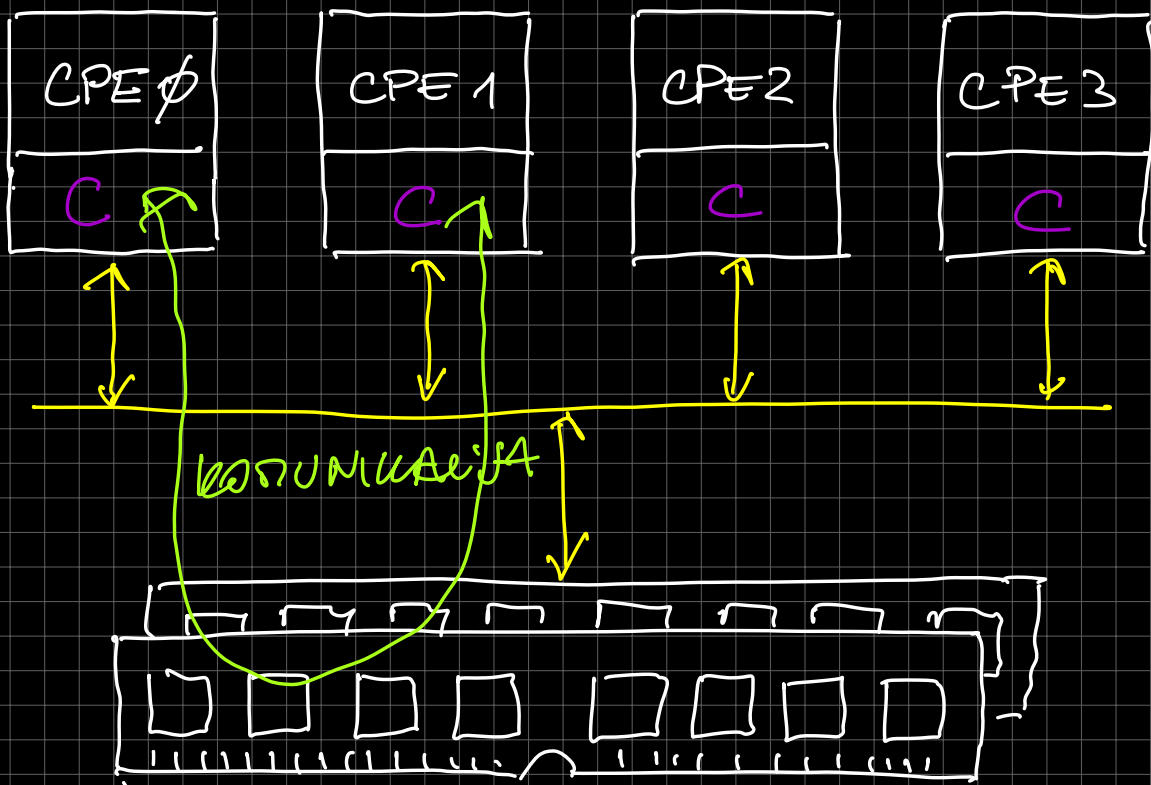


# PROGRAMIRANJE VEĆJEDRNIH SISTEMA S SLUŽBIM PORNILNIM



2x DDR4 SDRAM DIMM

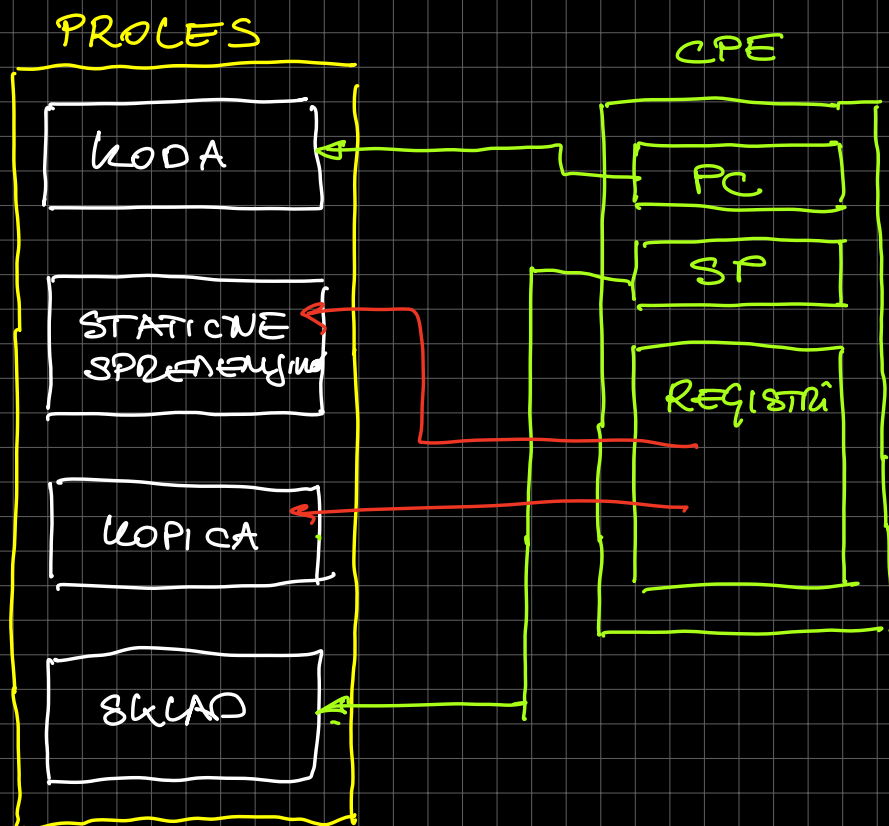
PROGRAMI IMAJU NEKI NEHABITET, KI  
NAN BO ONOGOLYL VPOREKNO IMAJANJE

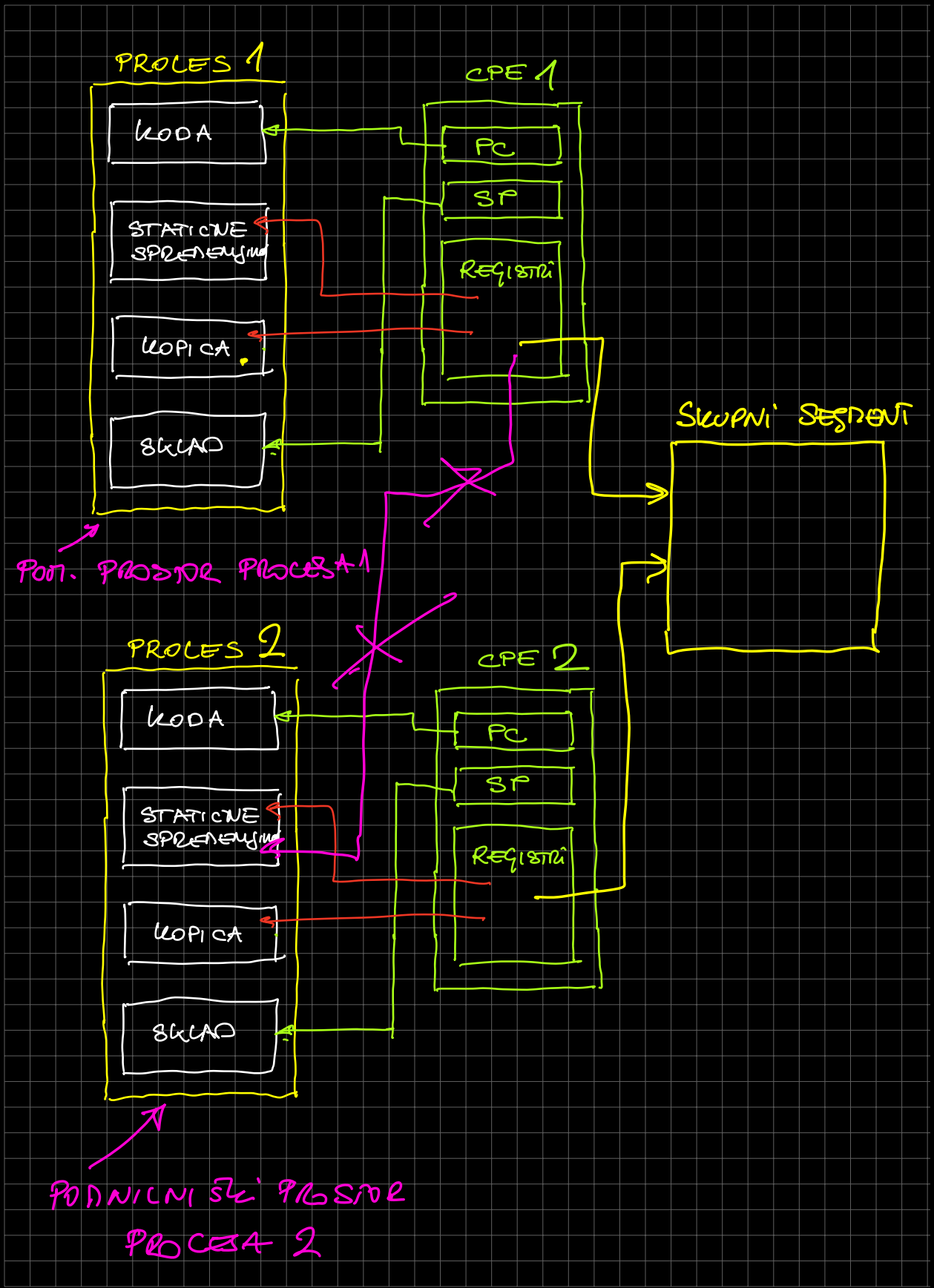
DEJAVJE PODROAV  
(KOMUNIKACIJA)

SINHEONITAEIJ

# 1. NAČIN PARALELIZACIJE: PROCESI

PROCES → program, ki ga OS upravlja  
z namenom izvrševanja





zjed: sestaveni elementu poja,  $2^{10}$  ma  
10<sup>6</sup> elements:

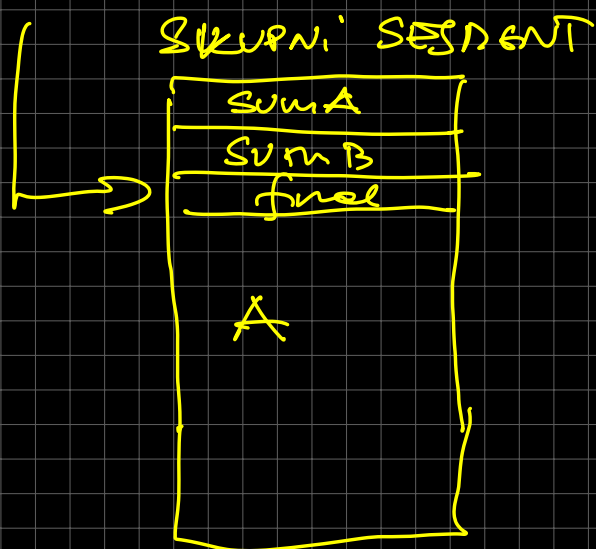
PROCES 1

PROCES 2

```
main ( ) {  
  for (i = 1 ... 500000) {  
    sumA += A[i];  
  }  
}
```

```
main ( ) {  
  for (i = 1 ... 500000)  
    sumB += A[i+500000];  
}
```

final = sumA + sumB



fork(), join(), share segment

# PROBLEM: RAZVRŠČANJE PROLESOV

PRELOP MED PROLES: ZAMTEVA  
VELIKO ČASA, KER JE TREBA  
SPRANITI CELOPEN KONTEKST  
NEKEGA PROCESA IN VNAJDI  
NOV KONTEKST IZ SPODINA

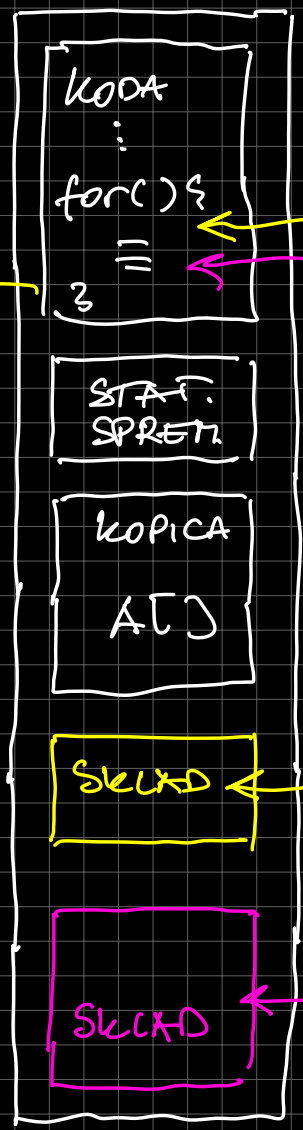
TAKAS SRE V NEKI DEL,  
KI SEŠA NE DA PARALELIZIRATI  
(CARDARIL)

IDEJA

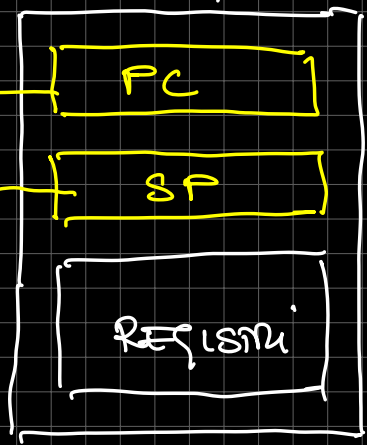
→ KAJ ČE BI SE PARALELNO  
AVDARJE DOJAZO  
ZNOTRAJ ENEGA PROCESA?

NITI

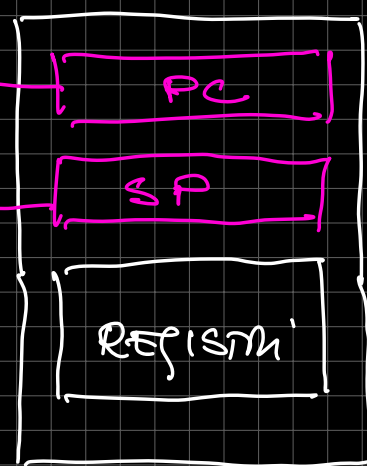
PROCES :



CPE 0



CPE 1



NASLOVNI PROSTOR  
PROCESA

NIT1 NIT2



TU MORAMO IMETI NOŽNOST, DA SE SPRADUJETA, ICASTERA NIT SMO

↓  
TO NAM PORA OPOVEDI OS ŽE NI  
KAK API



- Omgroca:

1. UPRAVLJANJE Ž NITMI = USVARJANJE  
IN USTAVJANJE  
*DANES*
2. SINHRONIZACIJA
3. MEĐSEBOJNO IZKLJUČEVANJE

USVARJANJE: `pthread_create()`

ČAKANJE  
NA KONEC: `pthread_join()`

