

Vzporedni in porazdeljeni sistemi in algoritmi, izpit

30. januar 2026

ime in priimek:

čas reševanja: 60 minut, naloge so enakovredne

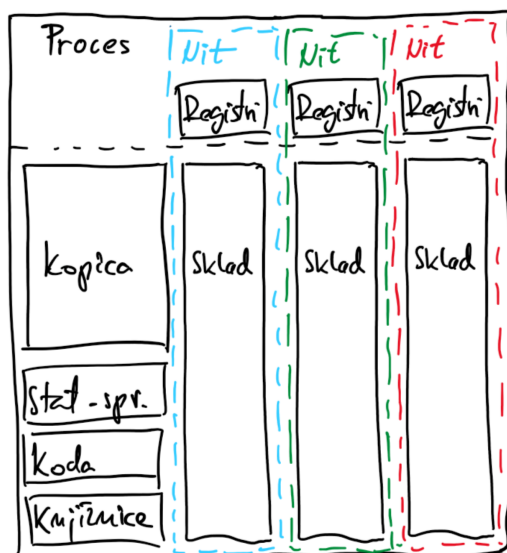
vpisna številka:

stran 1/4

1. Zakaj je arhitektura NUMA bolj raztegljiva (skalabilna) od arhitekture UMA?

- procesorska jedra so razdeljena v domene NUMA
- pomnilniški moduli so enakomerno razporejeni med domene NUMA
- dostopni časi do pomnilniških modulov so različni: krajši do pomnilniških modulov, neposredno povezanih na domeno, daljši do pomnilnikov bolj oddaljenih domen (časi so lahko do 3-krat daljši)
- izvajanje je učinkovito, če procesorska jedra iz izbrane domene čim več delajo s pomnilnikom, ki je neposredno povezan nanjo

2. Narišite podatkovne strukture večnitnega procesa v modernem operacijskem sistemu. Jasno označite strukture, ki so skupne vsem nitim, in strukture, ki so lastne posameznim nitim.



3. Kritični odsek zaklepamo z dvema ključavnicam, pri čemer ne moremo zagotoviti enakega vrstnega redna zaklepanja v vseh gorutinah. Napišite kodo v jeziku go za zaklepanje kritičnega odseka, ki preprečuje smrtni objem.

```
for {
    lockA.Lock()
    if !lockB.TryLock() {
        forkA.Unlock()
        time.Sleep(time.Millisecond)
        continue
    }
    break
}
```

4. V jeziku go želimo napisati program, v katerem glavna gorutina zažene N dodatnih gorutin, ki se izvajajo sočasno. Vsaka dodatna gorutina izpiše svojo oznako in zaključi. Program popravite tako, da se bodo vse oznake zagotovo pojavile na standardnem izhodu. Za sinhronizacijo gorutin uporabite kanal `ch`.

```
package main

import "fmt"

const N int = 5

var ch chan bool = make(chan bool)

func worker(id int) {
    fmt.Println(id)
    ch <- true
}

func main() {
    for i := range N {
        go worker(i)
    }

    for i = range N {
        <- ch
    }
}
```

5. Odjemalec je ob 9:15:11,023 po svoji uri strežniku NTP poslal sporočilo. Ob sprejemu sporočila je bila ura na strežniku 9:15:12,011, ob pošiljanju odgovora pa 9:15:12,017. Ob sprejemu odgovora je bila ura na odjemalcu 9:15:11,033. Na katero vrednost mora ob sprejemu odgovora odjemalec popraviti svojo uro? Čas je podan do milisekunde natančno.

$$\delta = (t_4 - t_1) - (t_3 - t_2) = 10 \text{ ms} - 6 \text{ ms} = 4 \text{ ms}$$

$$t_5 = t_3 + \frac{\delta}{2} = 9:15:12,019$$

6. V sistem za verižno replikacijo dodamo novo vozlišče. Kam ga dodamo? Kdaj začne novo vozlišče opravljati svojo funkcijo?

- *najprej uskladimo shrambo novega vozlišča z repom*
- *potem uredimo prevezavo tako, da rep postane vmesno vozlišče, novo vozlišče pa rep*
- *o naslovu novega repa obvestimo odjemalce*

7. V gručo za replikacijo z voditeljem (algoritem raft) je vključenih pet vozlišč. Zaradi težav z omrežjem gruča razpade na dva dela: v delu A sta voditelj in sledilec, v delu B pa trije sledilci. Kaj se bo v nadaljevanju ob zahtevah za vpis v dnevnik dogajalo v delu A in kaj v delu B?

- *A: zahteve za vpis se ne potrjujejo, saj so za potrditev potrebni trije; voditelj obvešča sledilca o svoji vlogi*
- *B: zaznajo da ni voditelja; izvedejo volitve; zahteve za vpis se potrjujejo (imamo tri procese kar je večina v pet-članski gruči raft).*

8. V gruči imamo več vozlišč, vsako vozlišče ima svojo repliko shrambe. Vsako vozlišče prejeto zahtevo za vpis v shrambo razširi na vsa ostala vozlišča. Sistem mora zagotavljati končno skladnost shramb. Če v shrambi uporabljamo ustrezne podatkovne tipe, si lahko privoščimo sočasno vpisovanje na različna vozlišča. Navedite nekaj tovrstnih podatkovnih tipov!

- *podatkovne tipe CRDT za replikacijo brez sporov (števcu, slovarji, množice, ...)*
- *monotone registre (ohranimo zapis z novejšim časovnim žigom, večvrednostni registri)*

9. Kaj v računalniškem sistemu virtualizirajo navidezni stroji in kaj vsebniki?

Nadzorniki virtualnih strojev virtualizirajo strojno opremo.

Nadzorniki vsebnikov virtualizirajo operacijski sistem gostitelja.

10. Kaj pri pripravi recepta za gradnjo vsebnika Apptainer napišemo v glavo in kaj v razdelke %files, %post, %environment, %runscript?

- glava: ime repozitorija in lokacija izvorne slike
- %files: datoteke, ki jih želimo iz gostitelja prenesti v vsebnik
- %post: izvajanje ukazov po vzpostavitvi vsebnika (nameščanje programske opreme)
- %environment: nastavitve okoljskih spremenljivk, ki se izvedejo ob zagonu vsebnika
- %runscript: ukazi, ki se izvedejo ob klicu apptainer run