

Vzporedni in porazdeljeni sistemi in algoritmi, izpit

9. februar 2026

ime in priimek:

vpisna številka:

čas reševanja: 60 minut, naloge so enakovredne

stran 1/4

1. Navedite glavne arhitekturne značilnosti sistemov s porazdeljenim pomnilnikom.

- vozlišča so povezana v omrežje
- vsako vozlišče ima ločen pomnilniški naslovni prostor
- komunikacija med vozlišči poteka z izmenjevanjem sporočil preko omrežja

2. Kako poteka razvrščanje gorutin na procesorska jedra?

- ob zagonu programa se ustvari zahtevano število niti, ki čakajo v bazenu niti
- program ustvarja gorutine, ki jih oddaja v bazen gorutin
- programski razvrščevalnik jezika go razvršča gorutine na proste niti
- razvrščevalnik operacijskega sistema razvršča niti na procesorska jedra

3. V jeziku go smo napisali program, v katerem glavna gorutina zažene več gorutin writer, ki spreminja globalno spremenljivko t, in več gorutin reader, ki spremenljivko t samo berejo. Kodo popravite tako, da pri sočasnem izvajanju več gorutin writer in reader ne bo prišlo do tveganega stanja. Poiščite učinkovito rešitev!

```
var t string
var m sync.RWMutex

func writer() {
    for {
        m.Lock()
        t = time.Now().String()
        m.Unlock()
    }
}

func reader() {
    for {
        m.RLock()
        fmt.Println(t)
        m.RUnlock()
    }
}
```

4. Spodnji program v go dopolnite tako, da bo nemudoma izpisal sporočilo, poslano v katerikoli kanal.

```
func main() {
    ch1 := make(chan string)
    ch2 := make(chan string)

    go func() {
        for {
            time.Sleep(300 * time.Millisecond)
            ch1 <- "VPSA"
        }
    }()

    go func() {
        for {
            time.Sleep(500 * time.Millisecond)
            ch2 <- "vpsa"
        }
    }()

    // manjkajoča koda
    for {
        select {
        case msg1 := <-ch1:
            fmt.Println("ch1:", msg1)
        case msg2 := <-ch2:
            fmt.Println("ch2:", msg2)
        }
    }
}
```

5. V porazdeljem sistemu imamo procese P1, P2 in P3. Vsak proces beleži interne dogodke in dogodke izmenjevanja sporočil. Dogodki na procesih si sledijo v naslednjem vrstnem redu: P1: a, b, c , P2: d, e, f, g in P3: h, i, j . Pari dogodkov $(e, b), (c, i)$ in (j, f) označujejo izmenjevanje sporočil – prvi dogodek v paru označuje pošiljanje, drugi pa sprejemanje sporočila. Določite vrednost vektorske ure ob dogodku g .

Vektorska ura ob dogodkih:

$a: [1, 0, 0], b: [2, 2, 0], c: [3, 2, 0]$
 $d: [0, 1, 0], e: [0, 2, 0], f: [3, 3, 3], g: [3, 4, 3]$
 $h: [0, 0, 1], i: [3, 2, 2], j: [3, 2, 3]$

6. V osnovnem sistemu za verižno replikacijo glava začne obdelovati naslednjo zahtevo za vpis šele potem, ko dobi potrditev o uspešnem vpisu od sosednjega vozlišča. Vpisovanje želimo pohitriti po principu cevovoda – glava začne obdelovati naslednjo zahtevo takoj potem, ko trenutno zahtevo za vpis preda sosednjemu vozlišču. Kako moramo dopolniti osnovno različico, da bodo bralci pri branju iz vseh vozlišč dobili enak odgovor? Namig: pisatelji lahko zaporedoma spreminjačo isti ključ.

Za vsak ključ dodamo verzijo, ki se ob vsakem novem vpisu poveča.

7. V gruču za replikacijo z voditeljem (algoritem raft) je vključenih pet procesov. Zaradi težav v omrežju gruča razпадa na dva dela: v delu A sta voditelj in sledilec, v delu B pa trije sledilci. V delu B izvolijo novega voditelja in odjemalci dodajo nekaj vnosov v dnevnik. Natančno opišite kaj se zgodi potem, ko so težave v omrežju odpravljene?

- *voditelj B pošlje razširja dnevniške zapise (ali srčni utrip) s kasnejšim volilnim obdobjem*
- *voditelj iz A zato sestopi in postane sledilec*
- *voditelj iz B sledilcem iz A pošlje manjkajoče dnevniške zapise*

8. Pri razširjanju sporočil aplikacija na procesu pošiljatelju sporočilo posreduje upravljalniku sporočil. Upravljalnik sporočil sporočilo posreduje vsem procesom v sistemu. Upravljalnik sporočil na vsakem procesu sprejeta sporočila na dogovorjeni način dostavi aplikaciji. Običajno upravljalnik sporočil sporočilo pošlje tudi samemu sebi, saj potem obdelava sporočila na vseh procesih poteka na enak način. Je še kakšen razlog za pošiljanje sporočila samemu sebi?

Zaradi zahtevanega vrstnega reda dostave sporočila v določenih situacijah ne smemo aplikaciji sporočila poslati takoj, ampak ga je potrebno zakasniti. Ker ne smemo blokirati pošiljanja sporočil, sporočilo pošlje tudi samemu sebi.

9. Narišite shemo virtualizacije z navideznimi stroji!



10. Kaj so prednosti virtualnih strojev pred vsebniki?

- virtualizirajo strojno opremo, namestimo lahko poljuben OS
- omogočajo interaktivno delo
- napaka v virtualnem stroju ne more zrušiti celotnega sistema
- nimamo težav z nezdružljivostjo gonilnikov in knjižnic v OS in vsebniku