

---

## Porazdeljeni sistemi

Izpit, 27. januar 2025, ob 8:30 v P22

---

1. Za sisteme s skupnim pomnilnikom sta značilni arhitekturi UMA in NUMA. Narišite shemi obeh arhitektur in podajte prednosti ene arhitekture pred drugo.

<https://github.com/laspp/PS-2024/blob/main/predavanja/02-arhitekture/arhitekture.md>,

Arhitektura UMA (SMP) in Arhitektura NUMA, UMA: enostavnost, NUMA: večja raztegljivost

2. Opišite povezavo med nitmi operacijskega sistema in nalogami (gorutinami v jeziku go) in kako se naloge izvajajo na strojni opremi.

<https://github.com/laspp/PS-2024/blob/main/predavanja/03-programska-oprema/programska-oprema.md>, Naloge: ko niti niso dovolj

3. V sočasnem programu se izvaja več nalog (gorutin v jeziku go). Kritične odseke kode varujemo s ključavnicami. Kdaj lahko pride do smetnega objema? Kako se smrtnemu objemu lahko izognemo?

<https://github.com/laspp/PS-2024/blob/main/predavanja/06-sinhronizacija-1/sinhronizacija-1.md>,

Hkratna uporaba več ključavnic in smrti objem (izpisani Coffmanovi pogoji, dva načina za preprečevanje)

4. S katerimi sinhronizacijskimi elementi bi lahko pripravili knjižnico za kanale v programskem jeziku, ki kanalov ne podpira? Napišite psevdo kodo za izvedbo kanala z navedenimi sinhronizacijskimi elementi. Omejite se na ključne dele kode, zasledujte sintakso jezika go.

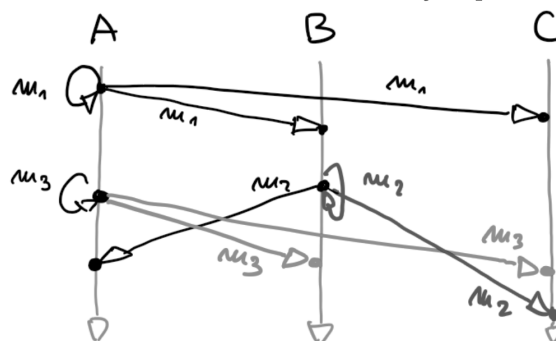
<https://github.com/laspp/PS-2024/blob/main/predavanja/08-sinhronizacija-3/koda/proizvajalci-porabniki-2.go>

5. Opišite protokol NTP. Kaj je ključna vsebina sporočil, kako po izmenjavi sporočil popravimo čas na odjemalcu?

<https://github.com/laspp/PS-2024/blob/main/predavanja/13-merjenje-casa/merjenje-casa.md>,

Usklajevanje ur po protokolu NTP, prenos podatkov in izračun odmika

6. Procesi  $A$ ,  $B$  in  $C$  razširjajo sporočila  $m_1$ ,  $m_2$  in  $m_3$  po spodnji shemi (čas narašča navzdol). S katerimi modeli dostave sporočil aplikaciji je shema skladna?



<https://github.com/laspp/PS-2024/blob/main/predavanja/14-razsirjanje-sporocil/razsirjanje-sporocil.md>, shema ustreza razširjanju FIFO in vzročnemu razširjanju

---

---

## Porazdeljeni sistemi

Izpit, 27. januar 2025, ob 8:30 v P22

---

7. V sistemu za verižno replikacijo je odpovedalo eno od srednjih vozlišč v verigi. Kako sistem zazna okvaro in kako ponovno vzpostavi verigo? Opišite, kaj se pri ponovnem vzpostavljanju verige dogaja s sporočili.

<https://github.com/laspp/PS-2024/blob/main/predavanja/15-replikacija-1/replikacija-1.md>, Verižna replikacija, nadzorna ravnina (delno), po prevezavi je potrebno poslati v eno smer sporočila, v drugo potrditve, na novo povezana procesa nadzorni ravnini sporočita zadnje stanje

8. Opišite postopek replikacije dnevnika pri algoritmu Raft.

<https://github.com/laspp/PS-2024/blob/main/predavanja/16-replikacija-2/replikacija-2.md>,  
Replikacija dnevnika

9. Grafični pospeševalniki imajo zaradi hierarhične arhitekture procesorjev in pomnilnikov v programskem modelu hierarhično organizirane tudi niti. Opišite prednosti hierarhične zasnove arhitekture in programskega modela.

Hierarhija prinese večjo lokalnost pomnilnikov, hitrejše dostopne čase do podatkov in s tem učinkovitejšo sinhronizacijo. Število niti v bloku, ki se lahko sinhronizirajo, je zaradi učinkovitosti omejena.

10. Imamo vektorja  $\mathbf{a}$  in  $\mathbf{b}$  dolžine  $n$ . Za grafični pospeševalnik pripravite ščepec, ki bo izračunal vsoto  $\mathbf{c} = \mathbf{a} + \mathbf{b}$  pravilno za vsako dolžino  $n$ , ne glede na organizacijo niti v glavnem programu.

```
__global__ void vectorAdd (float *c, const float *a, const float *b, int n) {  
    ...  
}
```

<https://github.com/laspp/PS-2024/blob/main/predavanja/21-cuda-primeri/koda/go/razlika/loceni-pomnilnik/razlika-4.cu>, v vrstici 13 odštevanje zamenjamo s seštevanjem

---