

Predavalnica: _____

Stolpec: _____

Vrsta: _____

Ime in priimek: _____

Vpisna številka: _____

Izpit pri predmetu
OPERACIJSKI SISTEMI
BVS-RI, FRI, UL

23. 6. 2021

Skupaj je možno zbrati **50** točk.

Čas pisanja: **75** minut.

Ta list **MORATE** oddati poleg pole z **imenom, priimkom in vpisno številko!**

Slikanje ipd. izpita pred koncem izpita je **prepovedano!** Izpit bo naknadno objavljen na eUčilnici!

Ugasnite mobitele! Prižgan mobilni telefon se smatra za poskus prepisovanja in se kaznuje z odvzemanjem izpita!

Literatura in kalkulatorji niso dovoljeni! Izjema je kalkulator z osnovnimi petimi operacijami: +, -, ×, /, \sqrt{x} !

Na poli označite naloge tako, da **zapišete številko naloge na levi rob lista, številko obkrožite ali napišite z drugo barvo (samo ne rdečo) ter naloge nanizate v enakem vrstnem redu kot na izpitu. Če določeno nalogo rešite na tem izpitu, to zapišite na poli ob ustrezno označeni nalogi.**

Izpit ima 14 vprašanj. Prosim, pišite **čitljivo**, pazite na slovnico!

1. (a) Naštejte glavne štiri funkcionalnosti (stebre) vsakega operacijskega sistema. V katero izmed njih spada algoritem princip ure? (1)
- (b) Kaj je ključna lastnost tega algoritma v primerjavi z ostalimi, ki rešujejo enak problem? (1)
2. (a) Kakšen problem ima zagotavljanje vzajemnega izključevanja z onemogočanjem prekinitvev? (1)
- (b) S katero drugo strojno rešitvijo smo rešili ta problem? (1)
- (c) Kaj pa je ključni problem te rešitve? (1)
- (d) In kako rešimo ta problem? (1)
3. (a) Kako izgleda zapis v tabeli strani (angl. *Page Table Entry*) pri navideznem pomnilniku? Razložite strukturo zapisa. (1)
- (b) Kakšna je ključna razlika te strukture v primerjavi s strukturo *PTE* pri enostavnem odstranjevanju? (1)
- (c) Če imamo 16-bitni logični naslov in pri tem prvih 6 bitov za naslov strani, kako sestavimo fizični naslov s pomočjo *PTE*? (1)
- (d) Kdo vzdržuje in kdo uporablja to tabelo? (1)
4. (a) Zapišite definicijo smrtnega objema. (1)
- (b) Eden izmed pristopov k reševanju smrtnega objema je tudi preprečevanje (angl. *prevention*). Kako preprečujemo (štiri) pogoje za potencialni in dejanski smrtni objem? (2)
5. Pri zamenjavi strani v glavnem pomnilniku smo vse naloge reševali s fiksnim številom rezidenčnih strani nekega procesa. Vendar smo obravnavali tudi spremenljivo število rezidenčnih strani – kaj veste o tem načinu zamenjave? (2)
6. (a) Ena glavnih podatkovnih struktur OS-a je *PCB*. Kaj so njegove (tri) ključne komponente? (1)
- (b) Kako se v Linuxu imenuje implementacija *PCB*-ja? (1)
- (c) Kako izgleda procesna slika v primeru, ko je funkcionalnost OS-a lahko del uporabniškega procesa? (1)
- (d) Zakaj je izvajanje OS-a znotraj uporabniškega procesa dobro? (1)
7. Kakšne so prednosti in slabosti niti na sistemskem nivoju (*KLT*)? (2)
8. (a) Kaj pravi Amdahllov zakon? (1)
- (b) Kaj ta zakon zanemari (česa ne upošteva)? (1)

9. (a) Kje smo najprej omenili problem notranje drobitve? (1)
- (b) Na primeru razumljivo pokažite koncept notranje drobitve. Skicirajte zaporedje korakov ter ga obrazložite. (1)
10. Načrtujemo sistem s šestimi procesi (P_1 - P_6). Ti bodo med izvajanjem potrebovali različne vire: R_1 - R_5 . Števnost virov v računalniškem sistemu bo: $R_1 - 3$, $R_2 - 2$, $R_3 - 2$, $R_4 - 1$, $R_5 - 2$. Potrebe so podane v naslednji matriki (v vrsticah so procesi, v stolpcih pa viri):

$$\begin{bmatrix} 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

V nekem trenutku ima P_1 v lasti dva vira R_1 , po en vir R_2 in R_3 , P_2 ima v lasti vir R_1 , P_4 ima v lasti R_4 , P_5 ima v lasti R_3 in R_5 , P_6 ima v lasti R_5 .

- (a) Narišite graf alokacije virov! (4)
- (b) Ali je sistem v smrtnem objemu? Obrazložite zakaj da, oziroma zakaj ne! (1)
11. Po postopku sistem prijatelj narišite razdeljevanje pomnilnika velikosti 256 MB za primer naslednjih zahtev procesov: A – 16 MB, B – 70 MB, sprosti A, C – 7 MB, sprosti B, D – 56 MB, sprosti C, E – 64 MB, sprosti D, sprosti E. V vsaki vrstici skice, ki predstavlja delovanje posameznega koraka algoritma, pazite na pravilno velikost celic. (5)
12. (a) Razporejevalnik dostopa do diska je dobil zahteve po sledih v naslednjem vrstnem redu: 340, 920, 930, 940, 950, 96, 380, 98, 101, 380. (2)
Kakšna je učinkovitost algoritma *SSTF*, če ima disk 1750 sledi in je na začetku glava diska na sledi 250?
- (b) Kakšna pa je učinkovitost algoritma *scan* v tem primeru, če se na začetku glava premika proti večjim sledem? (2)
- (c) Kateri algoritem je za ta tok zahtev boljši in zakaj? Pri tem tudi pravilno poimenujte izračunano metriko učinkovitosti! (1)
13. Imamo en procesor in več procesov, ki jih želimo izvajati na procesorju. Čas prispetja in čas izvajanja posameznega procesa je sledeč:

proces	čas prispetja	čas izvajanja
X	0	2
Y	1	3
Z	3	2
W	4	2
Q	5	1

- (a) Skicirajte kratkoročno razporejanje algoritma *feedback* pri $q = 1$ in treh prioritetnih vrstah. Zapišite podrobnosti postopka! (2)
- (b) Skicirajte kratkoročno razporejanje algoritma *SPN*. Zapišite podrobnosti postopka! (2)
- (c) Izračunajte učinkovitost obeh algoritmov in zapišite, kateri je boljši v tem primeru ter zakaj? Pri tem tudi pravilno poimenujte izračunano metriko učinkovitosti! (1)
14. V lupini Bash smo pognali `ls` in med drugim v izpisu opazili datoteko `check_running.py`.
- (a) S katerim ukazom bi prešteli število vrstic v datoteki? (1)
- (b) Kako bi pogledali, koliko prostora zavzame datoteka na disku? (1)
- (c) Kako bi pognali `check_running.py`, da bi ob morebitnem zaprtju terminala proces še vedno tekel? Upoštevajte, da trenutni imenik ni v spremenljivki `PATH`. (1)
- (d) Datoteka ima v osmiškem zapisu pravice 754. Zapišite pravice simbolično. (1)
- (e) Kakšna je razlika med izvajanjem `cat < check_running.py > python3` in `cat check_running.py | python3`? Bodite natančni! (1)