

Operacije: astma

Domaći jezik: (D 329)

UTORAK, 9-10 u → konzultacija

- TEKOVINE MODERNIH OS-a:

- višekadni rad (multitasking)
- višenamjenski rad
- upravljanje memorijom
- mehanizmi diskoncentracije
- kognitivne

→ pitanje za pravac u lečbi → BLOCI → minimum 5 blokova

↳ 10 blokova

→ 4 lekcije: detaljne pripreme na predavanju

→ 16 blokova

→ www.zemris.fer.hr / predmeti/los
D2 → 4 blok [upisati na vrijeme]

→ vježbe SC, SC i ŠC nisu obavezne, ali nate boljše

→ prvi je 1 blok (od 16) + odabrani laboratorijski

1. MI - 30

2. AVRIJU - 40

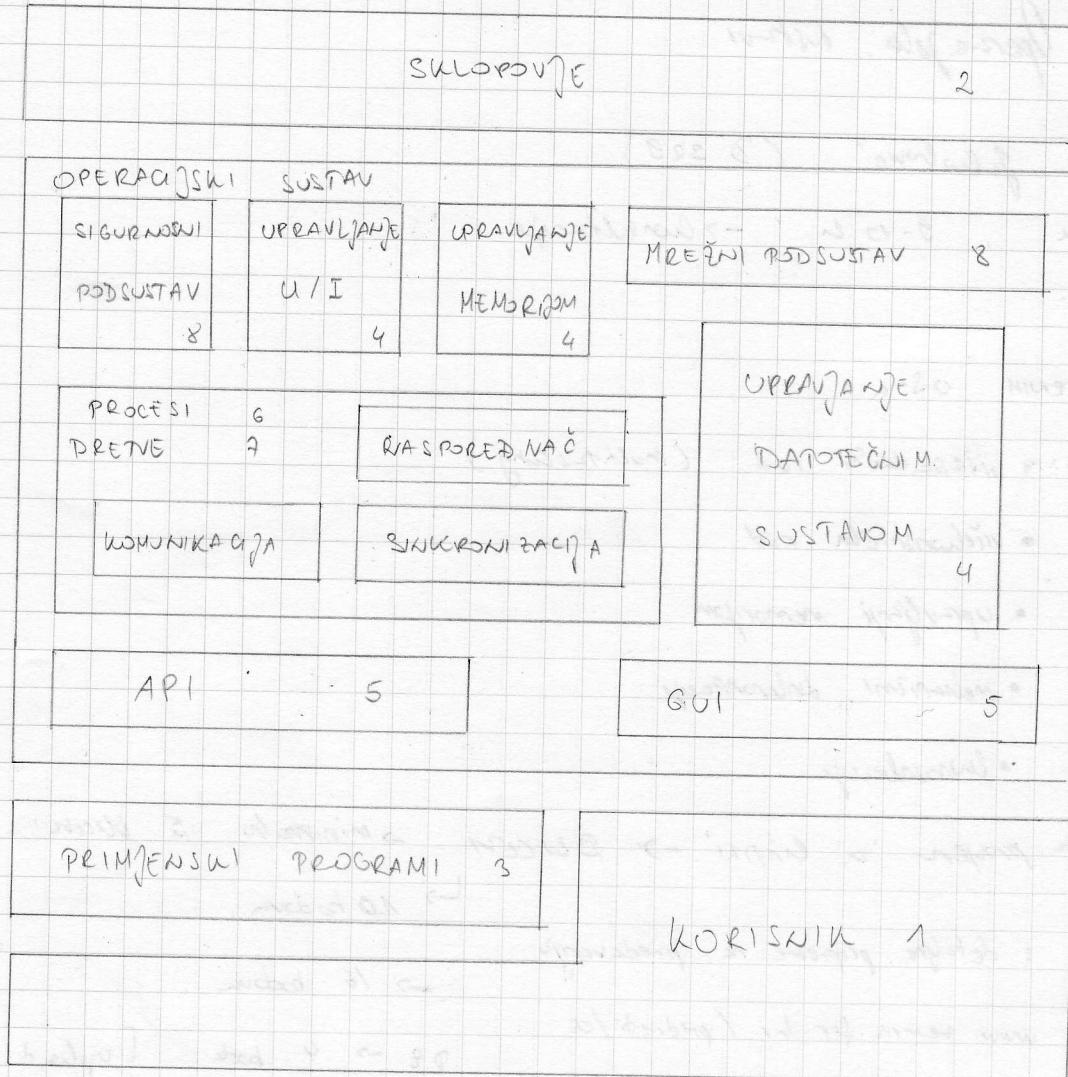
→ tako daću usmeni → blokovi, laboratorijski + usmeni → gvozdi komplikacije

↓
min 50%
lute mijenjaju

→ NE na ROK !!

SKLOPovanje

2



GUI → graficke korisničke sučelje

→ procesi ⇒ osnovni vistveni objekti rad

2. Računalo: model

SABIRNI CA

→ delovi:

- PODATKOVNI
- ADRESNI

- UPRAVLJACI

PROSEZOR

PC → adresa SLJEDEĆE naredbe

- ALU

- REGISTRI : PC, registri opće namjene, bezglaške stope (SP), register stope (SR)

- registri s neizvrsnim pristupom → instrukcija: register

↓
ne može im se
direktno pristupati

podatkovni medregistri

adresni redirektori

- upravljačka jedinica

while (1) {

obavet instrukcije s [PC]

PC ++

→ ne znam dačko PC++

delusirajuća instrukciju

ide učinak delusirajuće

obavet instrukciju

y

Instrukcije direkte

Jeden instrukcija

- spremnik i procesor te vodio odgovor za izvođenje programa
+ vrednost
- na instrukciju koje procesor računa naziva se direktiva
- program \Rightarrow na instrukcije u spremniku (oktalin)
- direktiva \Rightarrow na instrukcije koje te vode u vreme (dynamički)

Proces

- program koji je izvoditi (ne sto je potreba da bude program niz ili redoslijed)
- izvođenje procesa uključuje izvođenje skrenutih direktiva, more ih biti i više (ako ima više procesora)

2.1. Sljep registar i sljep procesor instrukcija (?)

2.3. Jedinica pomoći spremnik koji će ostati u vremenu (1 proces sa strane)

2.5. Kontrolni registar direktive ili stop register (SR i PC je zatvoren direktno) sa strane

procesni adresni prostor



vise direktnih adresnih prostora

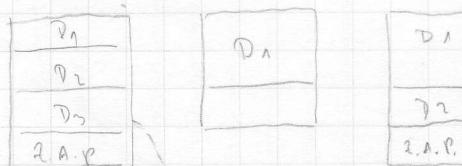


instrukcije, lokaci + stop (3 direkta)

P1

P2

P3



\Rightarrow mehanizam sumnje kontroli

osigavljanje vise direktnih red

\rightarrow direktive daju vremena koje ne obavljaju (npr. IJ - jedinicne)

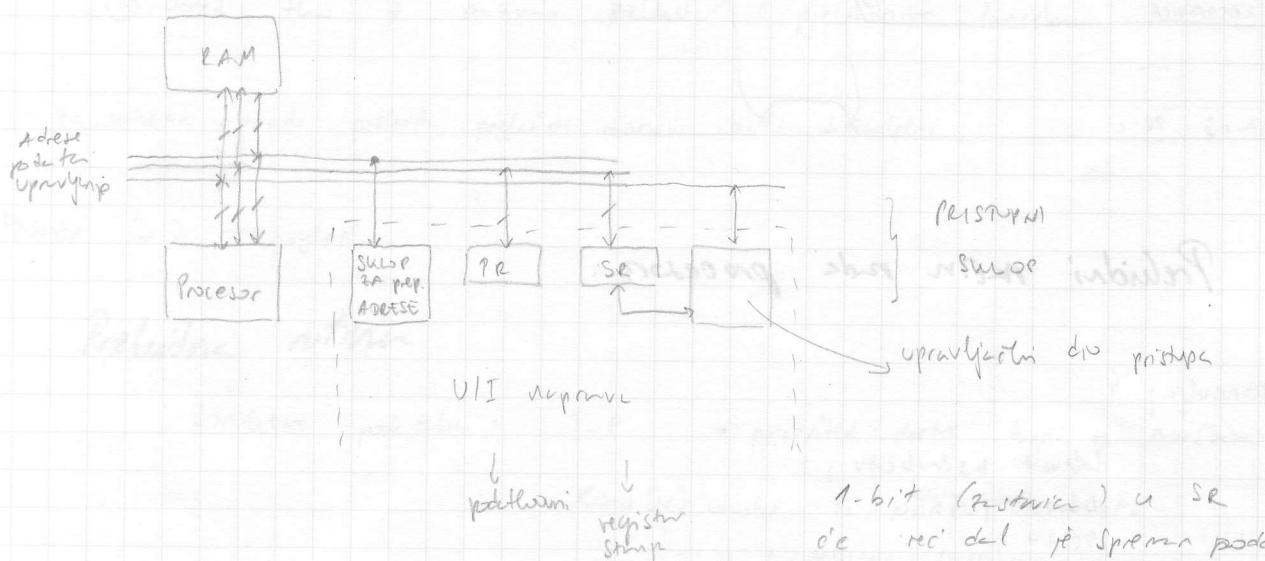
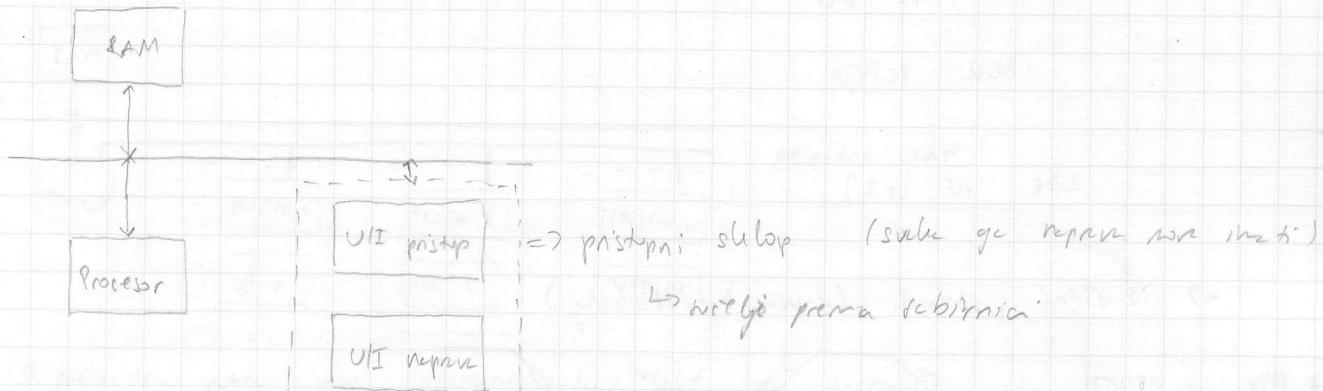
\downarrow
nogovi okupljeni druge direktive



3. Obavljanje UII operacija, preludni rad

2 osnovne tipa pristupa memoriji:

- 1) zasebno po zasebno (stupljavanje)
- 2) blok po bloku



INSTRUKCIJE:

LDR r0, [r1] → adresu podatka fi u RA

STR r0, [r1] → pohrani r0 na adresu fi u policijskoj r1

ADR r1, SIMPIME → u r1 smjesti adresu labele

CMP r0, #0 → usporjava register s brojem

BEQ PC=RA → skok

BAL PC=A → branch always

• prenosi se zadržava:

→ norma podatka

du je ($r = \phi$); // nema očekivanje

površina PR;

ADD r1, RS (register stamp)

ADD r2, PR

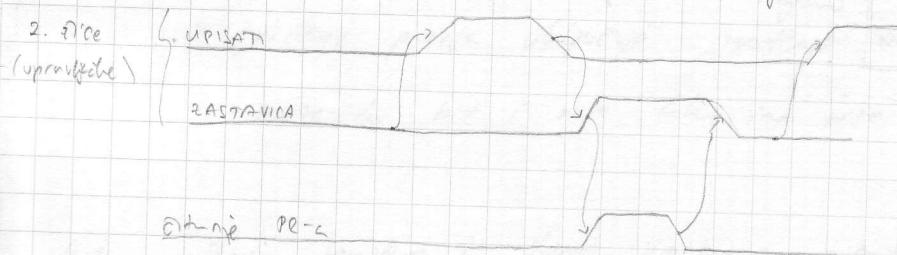
PETRYA : LDR r0, [r1]

CMP r0, #0

BEQ PETRYA

LDR r0, [r2]

⇒ 2-zidni protokol (2-wire handshaking)



Prelidni način rada procedura

Ponavljajući

doljni instrukciju;

deksplodirati;

PC++;

odredi gdje je operandi...;

predi naredbu;

početi rezultat;

also je postavljen prelidi signal i prelidi su dozvoljeni }

1. taborni način prelida

2. projekti u sustavski način rada

2A. aktiviraju sustavsku licenziju stoga

2B. adresiraju sustavski adresi pristor

3. PC i RS početi na sustavski stog

4. PC \leftarrow adresu rutne za obranu prelida

↓
relacijski u procesoru (skrivni register)

- učivo obrade preluka, prvo stek obnovit kontekst (bp, PC i SR)

pa se vrati u preluku dretu

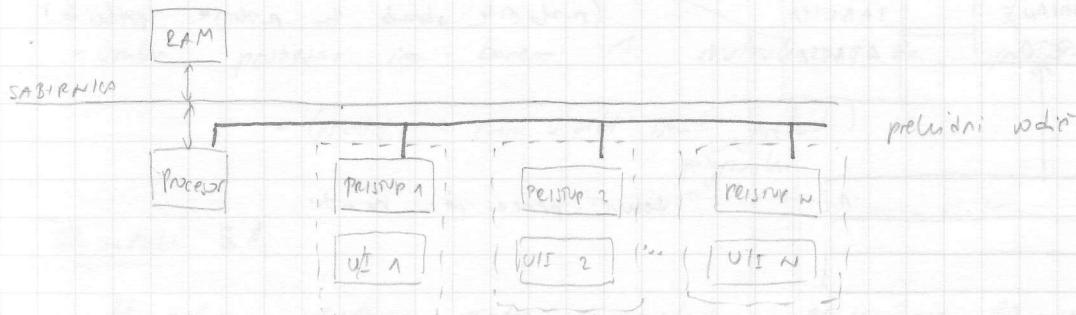
RS = SR

= obnovi RS i PC

→ automatizirani projekt u novom reču prelukute dretre (obnovi SR)

→ dozvoli preluku

→ ne mogu vrati PC



① prelukne rutine prvo počevši kontekst na sastavljeni stup

② odrediti kada je razvran preluk : preluknim lancem

! - za vrijeme obrade preluka, preluki moraju biti dozvoljeni

Primer 3.3. → pogled

Prelukna rutina

Struktura podataka : T-P → prioritet dretre koja će uranjati

Kon[N] → polje za phon konteksta
→ N kontekstova (U/I jedinic)

K-2[N] → polje rastavice svih konteksta

? → pogledaj problem PPS?

- ako prelukne 5 preluke 3, kontekst 3 se spremi u Kon[5]

? Primer 3.4

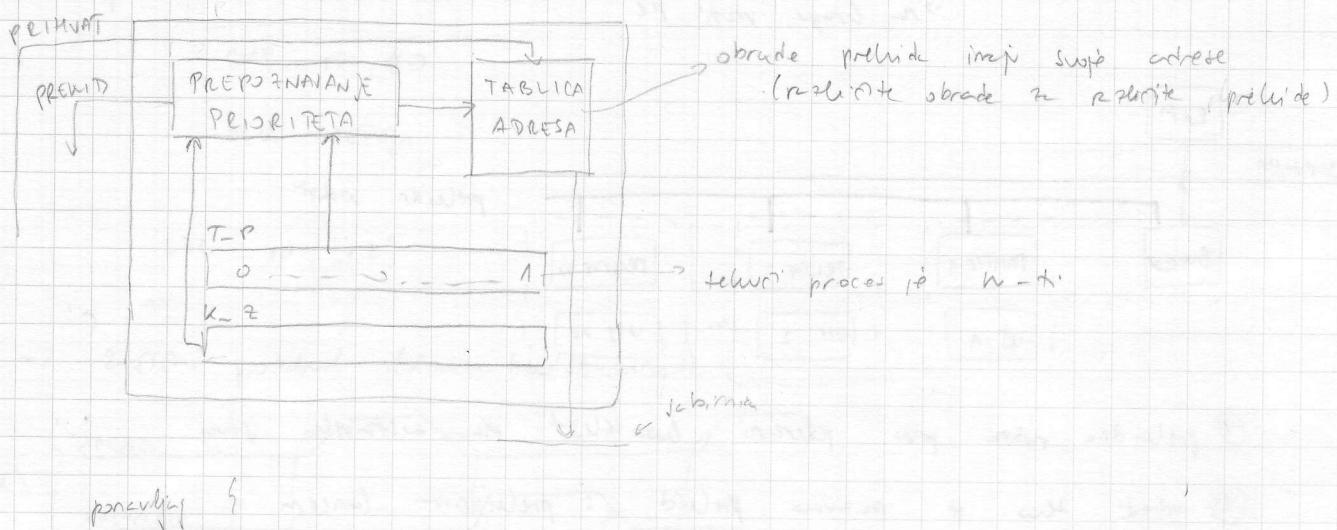
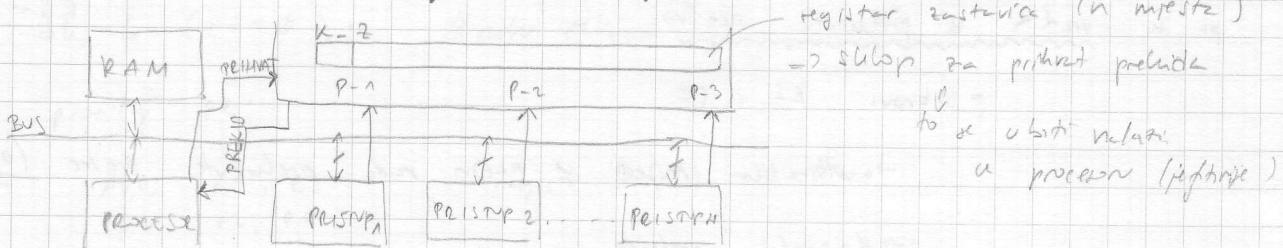
korisnicka dretva ima prioritet 0 (najniži)

→ prelukna rutina troši dosta vremena

↳ rješenje : ulogu za prihvatanje preluka

⇒ prvi kontekst stavi u stup, pa onda u kon[] ⇒ dupli putan

Slučaj posluša potporu prekidom podstavka



povratljivo

akko je (prekidi signal postavljen i dozvoljen je prekidanje)?

- ① zaborani prekidanje
- ② priček u sustavskim radom
- ③ PC, ES i sve ostale registre (kompletan CounterList)

zaborani na sustavsku liniju

- ④ postavi signal PREKIDAT
- ⑤ $PC \leftarrow$ adresu s adresog dijela kibernine

⇒ processor će učiniti "čuvanje poslova" samo za one prekide koji su bitni

⇒ one čije slučaj "propasti" (vrijednost ne dolazi do procesora)

Preludi generizni avtor procesora

• uveljavlja se program
• dobitnik podatkov je
prelud
• uveljavlja se
program
• funkcija

- u vseh izvajanjih delujejocega procesorja mora obrediti preludi:
 - dejavnosti s nizom
 - delovanje repozicije instrukcije
 - adresiranje repozicije adreseve dolucevje

- skupin' procesator ima barem 1 instrukcijo za programski preludi
 - (print. → mora vsebiti imati preludi)

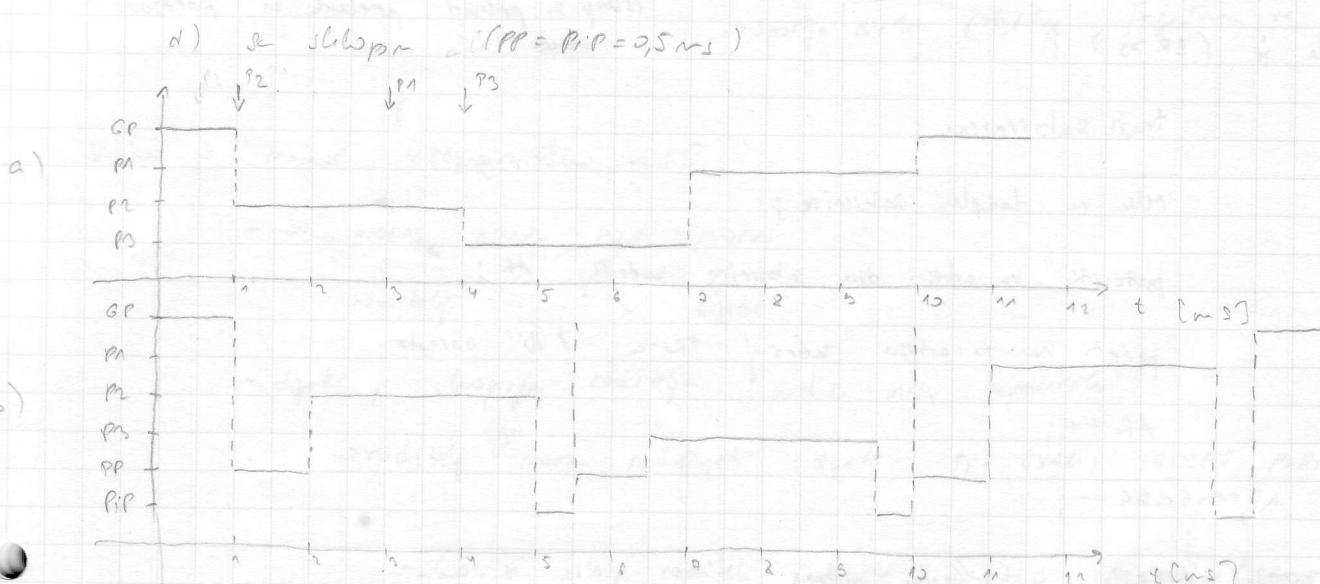
Zadatak 3.1.

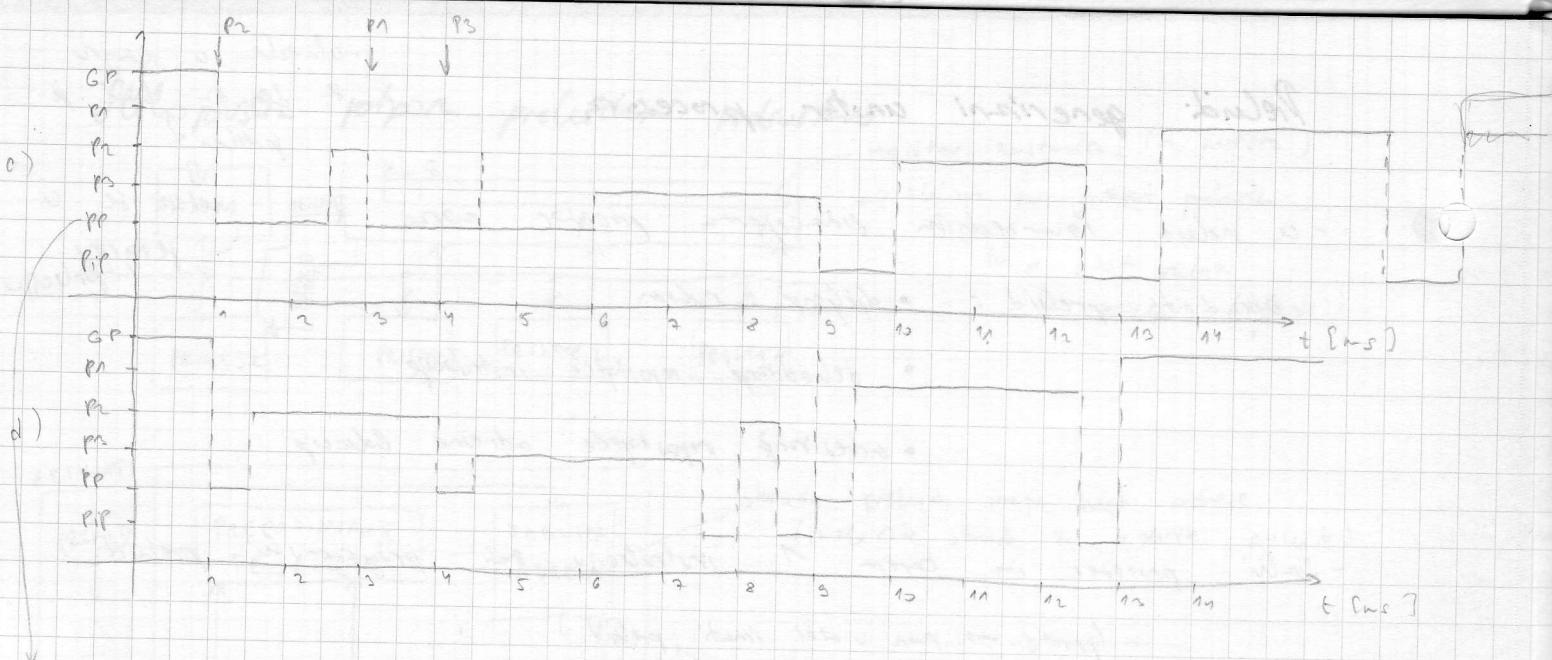
U vseh treh različnih delovanjih procesorja P1 u 3ms, P2 u 1ms, P3 u 4ms.

Priroditelj preludi odredita je brojem ($P_3 > P_2 > P_1$). Obrači ^{svetlobno} preludi dobre

$P = 3\text{ ms}$. Grafiki prikazati aktivnost procesora u glavnem programu (GP) procedurami in obradi preludi (P_i) te procedurami in prihvat preludi (P_{iP}) i povratki iz preludi (P_{iP}^R) i to:

- a) u rednem sledjenju
- b) bez sklopa za prihvat preludi (prihvat dobre 1ms, $P_{iP} = 0,5\text{ ms}$)
bez reaktivne rutine (obrači iz zahajenih preludov)
- c) bez sklopa ali s programskim potporom ($P_P = 1,5\text{ ms}$, $P_{iP} = 1\text{ ms}$)
- d) s sklopmi ($P_P = P_{iP} = 0,5\text{ ms}$)

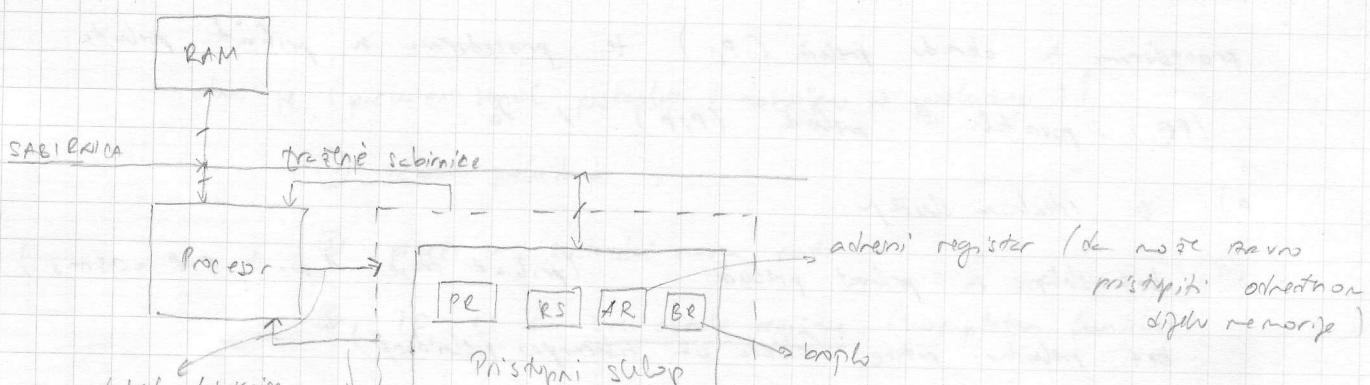




over PP i me snížtu → stavi kontext na stej blok blok → aby se ujistil, že je v něj překlán → obnov kontext

PiP → obnov kontextu + obnovit s jistotou, že

Príjem blokov znaku, DMA



ale je (BR >) ; (slap je prikaz prekida v procesor)

tržníčku;

alec na dodatkú sabirnicu;

postavti na adresni diu sabirnice sadrží tři;

prekici na tu adresu sadrží pravé, ktoré obratne

AR ++;

BR --;

postavti PREKID ; //gotov prenos

→ TMA & solo prime o tom, keď de ujet sabirnica

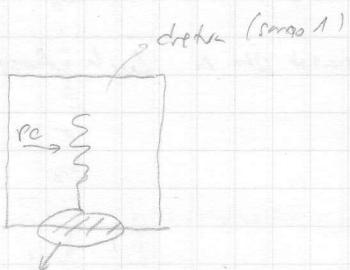
→ sklop je prihvati preljeće ce ne smjer signal na skladištu skidat učinku
za rutnu logiku bude da je prihvata sprem

4. Međusobno ističuvanje u više dretvama

1 3 5

Ljstavina

Proces nekada



virtuelni procesor
(lightweight proces - LWP)

danas



a-a

→ može se ograničiti da u dretvi budu
samo 1 proces

→ solj : viseprogramski rad (paralelni (n>1) procesi)

→ procesi tekuju konzistentno → operativni sustav im neda (jer se neki delovi
smeđi međusobno) ⇒ postoji dretve

• novaj nisk
zadnjicima → podjeli osi postor

→ dretvom je sve zadan (dretvom ište proces, i odc.)

↳ za fiksne adresne prostore (globale varijable)

→ tako se dretve istovremeno podeljuju $a=15$ $a=16$

⇒ sprječiti dretvu problem (izjaviti se dinamizacija)

Ukles je moguć višeprogramski rad?

- 1 proces stvara novi proces
↓ ↓
roditelj djete

- djete je kopija roditelja (postoje novi zadnjici)

→ roditelj mora nadgrijeti djete tj. SVAKI PROCES MORA IMATI
RODITELJA

→ tako se ubije roditelj, djete postaje orphelin (smrť)

o upravo se mora brisati
OS

process 1

int fork (wid);

↳ stvara novi proces (dvojno raditeљe)

→ A (nem može se novi proces) Error

→ vrati 2 integera ?? :

• PID datch → proces ID (identifikacija broj)

• 0 → datch

TID - thread ID

grš arguments

✓ per man vrati vrijednost i dodataku (očekivani polazak isto ne funkcije)

→ switch funkcija se mora koristit

ZOMBIE → proces koji je završio s zwrtjenjem ali neye reapis po postupku

u publicanu osz

Vježba 6.1.1. red

DRETVI

ULAZNA

	A	B	C	D	...
--	---	---	---	---	-----

RAZINA

	A	B	C	...
--	---	---	---	-----

IZLAZNA

	A	B	...
--	---	---	-----

- kritični odjeljci \Rightarrow dretve istovremeno otvori isti podatki

- jednina funkcije u svim mrežama stoji u skladu (visoku paralelnost)

- dretve se diskretniziraju pravom jedninanom funkciju

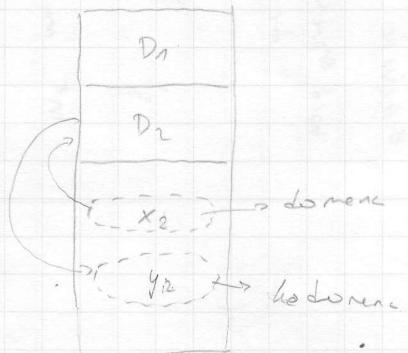
- rukovanje podatkovima os-a \Rightarrow kritični odjeljci za jedninanu funkciju

- Slučaj nem. kolonija \Rightarrow kroz dretve iste mreže se domen
- Slučaj nem. kolonija \Rightarrow kroz dretve nisu podatke
je kodiran.

$$f_2 : x_2 \rightarrow y_2 \quad f_1 : x_1 \rightarrow y_1$$

UVJET NEZAVISNOSTI ZADATKA

lobavljaj ga dretve



$$\begin{matrix} D_i & \in & D_j \\ x_i & \rightarrow & y_i \\ x'_i & \rightarrow & y'_i \end{matrix}$$

$$(x_i \cap y_i) \cup (y_i \cap x'_i) \cup (y'_i \cap y_i) = \emptyset \quad \Rightarrow \text{zad. su dretve zavisne}$$

presek

Zadatak 4.1.

Ljekov zadatak je zadan:

$$z_1 \rightarrow z_2 \rightarrow z_3 \rightarrow z_4 \rightarrow z_5 \rightarrow z_6 \rightarrow z_7$$

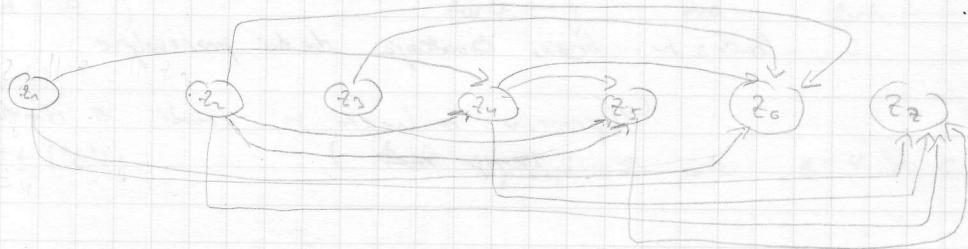
Zadatak je raspodeljen na 6 mesta (M₁ do M₆) prema tablici.

	z_1	z_2	z_3	z_4	z_5	z_6	z_7
M ₁	D	D	D	D		K	
M ₂	K			K		D	D
M ₃		K		D	D, K		K
M ₄			K		D		
M ₅			K			K	
M ₆							

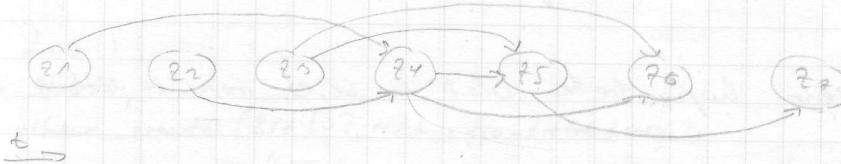
Slučaj
mesta
loševje

Otkazati minimalno-paralelni ljekov zadatak u kampu u obzir
upisov redoslijedi odvoj u lancu.

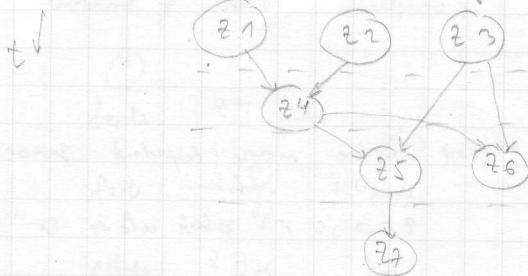
①



② → uklanjanju redundantnih tavljnosti



③



Medusobno isključivanje

abu (1) ;

K.O. ;



N.K.O. ;

}

→ 2. definicija rešenja medusobno biti u kritičnom određenju

→ Uvjeti kada je algoritam med. isključivanje npr zadovoljstvi:

0) direkte izlazne K.O. medusobno isključuju

1) mehanizam medusobnog isključivanja npr. odeljivanje u grupe

kada u brane strategije detekcije prioritete

↳ (procesor na kojem se izvodi npr. direktne mesečne
funkcije radi)

2) kada nema od detektivskih rezultata u N.K.O., što ne
smije dovesti drugu detektivu da uđe u K.O.

3) kada kada detektiv u K.O. npr. se oduže u
branama unesenih

0) (kada nema rješenje)

dole je (1) ;

dole je ($z = 1$) ;

→ objekti detektive mogu se podružiti povezani

$z = 1$;

$z = 0$ i objekti u K.O.

K.O. ;

$z = 0$;

N.K.O. ;

y

① for (i=0 ; i<5 ; i++) {

if (fork() == 0) {

→ start de 5 process

process (&t);

exit(0);

y
y

②

fork();

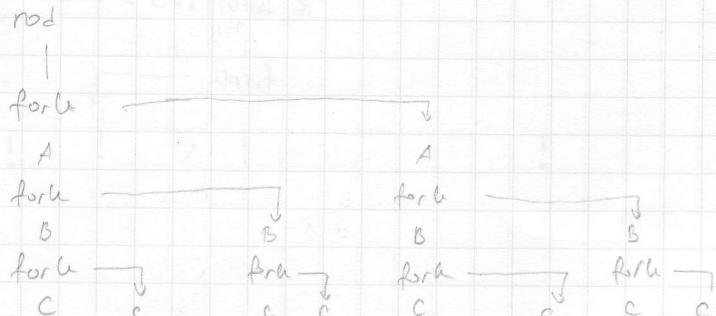
printf ("A ");

fork();

printf ("B ");

fork();

printf ("C ");



A se inscrije 2 puturi, B 4 și C 8 puturi

! → după fiecare fork se modifica status

③ fork();

fork();

pthread_create (&id[0], NULL, fra_dreptare, &arg);

- // -

Unde și când se returnă?

①

fork

|

fork

|

pthread

|

pthread

|

pthread

②

fork

|

pthread

|

pthread

|

pthread

③

fork

|

pthread

|

pthread

|

pthread

④

fork

|

pthread

|

pthread

|

pthread

12 drepturi ulungos

④ if (fork() == 0)

if (fork() != 0)

fork();

printf ("! ");

natély

if fork() == 0

if fork() != 0

fork

!

!

!

!

$\Rightarrow 4 \text{ işpiş}$

∴ nesståvka

Druži polusci (ang. algorithm):

PRAW = 0 (0' 1

\rightarrow Cwycia dretka smpie 101' K.O.

do { } ;

do { } (PRAW ==]);

K.O.

Udretva] je u le. o.

PRAW ==]

NiK.O. ;

y

\rightarrow napričko je] predugo u NiK.O. \rightarrow] može 2. put zeli u K.O. ali nemre!

Treść polaryzacji:

int z[2];

do u (1) {

 do u p (z[i] == 1) {

 z[i] = 1;

 k.o.

 z[i] = 0;

 k.o.;

y

 return;

int z[2];

do u (1) {

 z[i] = 1;

 do u p (z[i] == 1);

 k.o.;

 z[i] = 0;

 k.o.;

y

Pet. polaryzacji:

do u p (1) {

 z[i] = 1;

 do u p (z[i] == 1) { z[i] = 0;

 do u p (z[i] == 1); // odkreśl

 z[i] = 1;

y

 k.o.;

 z[i] = 0;

 k.o.;

y

mechanizm
wykrycia

Test polaryzacji (Delekten wypalen)

```
int z[2]; praw;
druk(i) {
```

$$z[i] = 1;$$

```
druk (z[i] == i) {
```

```
    else if (praw == i) {
```

$$z[i] = 0;$$

```
druk (praw == i); // zdroj odklape
```

$$z[i] = 1;$$

← K.O. :

$$praw = i;$$

$$z[i] \neq 0;$$

N.K.O.

] bitem redagowanym (typ, ale
zasupe)

Peterson:

```
druk (i) {
```

$$z[i] = 1;$$

→ praw = j;

```
druk ((prawoc == j) && (ZASTAVICA[j] == 1));
```

K.O.;

$$z[i] = 0;$$

N.K.O.

}

⇒ peterson nie kłopid delekten nie kłopid w dekre s mazurin brzegiem

also 2 istotne digne zastavica ⇒ sprawa deje praw brzeg

⇒ preferuje brzeg dekretu

Leba - Lamportov algoritam međusobno isključivanje (baker's algorithm)

Broj

- u dretu' → svaki dobije svoj broj

int BROJ [n];

int ULAZ [n];

dok (1) {

ULAZ[i] = 1;
čitati ZADNJI_BROJ;
BROJ[i] = ZADNJI_BROJ + 1;
ZADNJI_BROJ = BROJ[i];
ULAZ[i] = 0;

ulač broj vrati

za (j = 0; j < n; j++) {

čitati varijablu ULAZ[j];

dok (ULAZ[j] == 1) {

čitati varijablu ULAZ[j];

various

T

prihvati aho
dretva J

uprav ulaz
croz vrati

pregleđivanje

svih dretvi

čitati varijablu BROJ[j];

dok je (BROJ[j] != 0 & (BROJ[j], j) < (BROJ[i], i)) {

čitati varijablu BROJ[j]

TID

K.O.;

BROJ[i] = 0;

N.K.O.;

akcija se slijedi u brojivo
isti => onda ona s
nigim TID-om dobije
red

prve mjesto

→ u svim ovim algoritmicima postoji redoslijed

|| → dretvi
mora čekat svoj
red (dok ona
ne bude dretva s
najmanjim brojem)

Slabpowstanie poprzez mediarzem i synchronizacją

→ IN-KLUSIVE NARĘCZIE:

TAS (test and set)

SWAP

→ za wiele przeszkód jest wąska

→ niedostępny adres

FETCH-AND-ADD

→ po głosach kiedyś do funkcji razy' zapisz R.O.

CAS

(compare and set/swap) → CAS (adres, usparec, postawi):

/* 1. zapisz nowy adres */

prostros = obiekt - s - adres (adres);

also je (prostros == usparec) {

/* drugi zapisz nowy adres */

postawi - nowy - adres (adres, postawi);

y

dobra :

dobr (1) ;

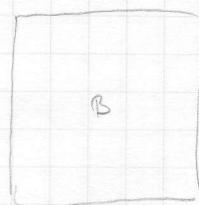
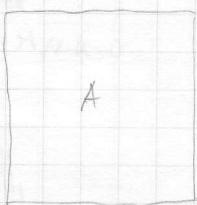
dobr : (TAS(z) == 1); /* zmaia wartość z 1, ale unikaj postawi z u 1

K.O.;

z = 0;

N.K.O.;

Zapewnienie pamięci (2 proces)



2. ADD.
PROSTOR

→ daje możliwość (size)

→ zmienić (size)

→ daje możliwość (size)

→ get user id () → jednostki bryl'

zakres typu
0 600

- flags → prawo przypisania

0 1,6 0 0
E W X --- ---
user group other

→ za datę taka

shmatc \Rightarrow funkcije za razdvajanje memorije procesora A i B

shmatc \rightarrow brije učenje s procesom

? shmatc \rightarrow osloboditi se memorije \Rightarrow da treba napraviti nešto

shmatc \rightarrow brije se memorije

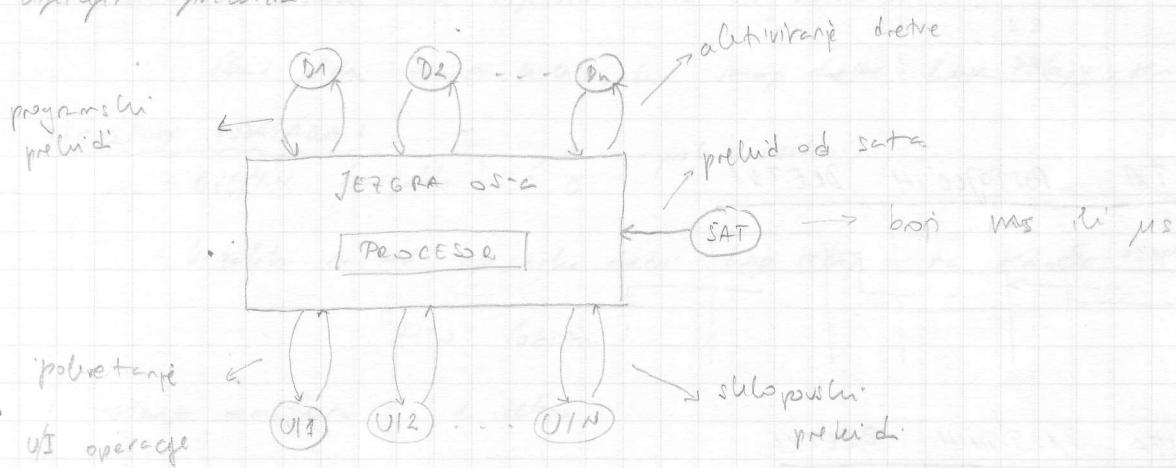
5. Jezgr OS-a

- jezgr OS-a je sklop od:

• struktura podataka regre

• program funkcija (f-fc)

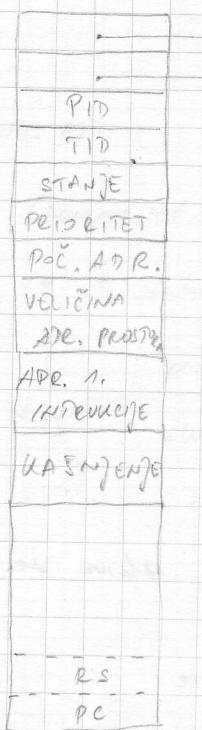
- uklanjanje jezgra može posivi program funkcije sa zbiru se pod operacijem prelaska



- izlazak iz jezgre može se na polje za operacije redne od dretvi

Struktura potreba reprezentacija

Osnovne dretve



jer je
fiksne liste

→ aktivna / neaktivna
→ podaci se raspoređuju dretvi, način raspoređivanja

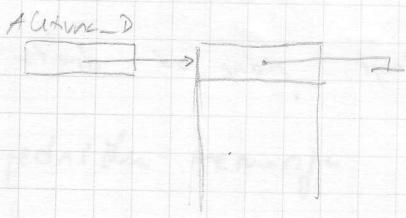
→ sleep

PROSTOR za kontekst

LISTA POSTROJECIH DRETVI



LISTA AKTIVNIH DRETVI



→ svaki proces ima svoju listu

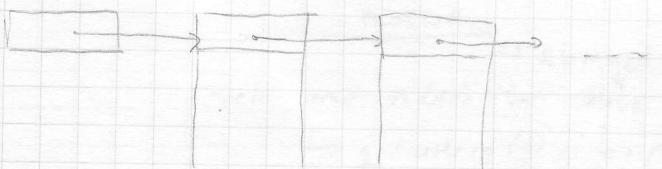
(svaka je dretva u časti)

↑
Označava se redoslijed u procesoru

→ listički na slaganje u opisnik prethodne dretve

LISTA PRIPRAVNIH DRETVI

Pripravne - D



(FIFO, se pothodi)

Blokirajuća stanja detri

- detri na: • binarnim semaforu (BSEM)
- operacijem semaforu (OS, i. OSEM)
- da bazu čekanje (sleep)
- detanje na savršenih UI operacija

SEMAFORI

BSEM

- program međusobno buji drugi za funkcioniranje detri
- desetični kod strukture potrebna je programski fija
- "izlaz" → kada bud naveden detri ne očkuje

(kada je detri rezerv u. o., tada imaju detri buji očekju, tenu postupe
"izlaz")

STRUKTURA POZATAKA:

- BSEM.v (variable) \leftarrow $\begin{cases} 1 & -\text{proba je zelen} \\ 0 & \end{cases}$
- Detalje na red operacija detri: buji očekju na semafor
 - ↓ FIFO (obično)

→ stanje semafora : 1. zelen

- 2. crveno:
 - nemu mlijeva u redu
 - ima mlijeva u redu

IZVJEŠĆE FUNKCIJE

- ispitaj-BSEM ili očekj-BSEM (postavi je prije u. o.)
- postavi-BSEM

dok 11/3

čekaj-BSEM (i);
u. o.;

postavi-BSEM (i);

u. u.;

y

ULAZNA

DRENA:

dna (1) {

delij - podatki (1);
delij - BSEM (1);
delij - podatki - nivoj - drevo (1);
postavi - BSEM (2);

W.O.

Non. steve "nivoj" na zbroju,
 $v = 0$;
Nedost upisa v isti podatku

}

BSEM (1). $v = 1$;
BSEM (2). $v = 0$;

Napetina napredost

RADNA DRENA:

dna (1) {

delij - BSEM (2);
primi - podatki (1);
postavi - BSEM (1);
obradji - podatki (1);
delaj - BSEM (3);
selj - podatki (1);
postavi - BSEM (4);

}

BSEM (3). $v = 1$;
BSEM (4). $v = 0$;

ULAZNA DRENA:

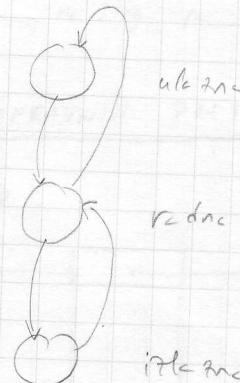
dna (1) {

delaj - BSEM (4);
primi - podatki (1);
postavi - BSEM (3);
prilazi - podatki (1);

W.O.

}

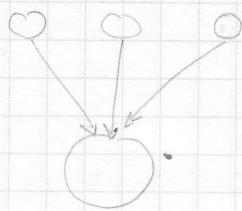
-sem nem redog obvezje, am i regulira rezultiraju izvodenja dreva



Opcii semafor

⇒ jezgrije mehanizam korišćen u

sinhronizaciji dretva



⇒ može imati sinhronizacijski mehanizam za blokiranje dvega dretva

• OS (Djelatnost OS)

strukture podataka:

① OS.V : $(-\infty, \infty)$

$[1, \infty)$ - zelen

$(-\infty, 0]$ - nezelen

② konzgle

i-fre:

delaj - OS(1);
postavi - OS(1);

• OSEM (objektni semaf)

strukture podataka:

① OSEM.V : $[1, MAX-VR]$ - zelen

0 - crven

② konzgle

i-fre:

delaj - OSEM(1);
postavi - OSEM(1);

Napoci stanje dretvi :

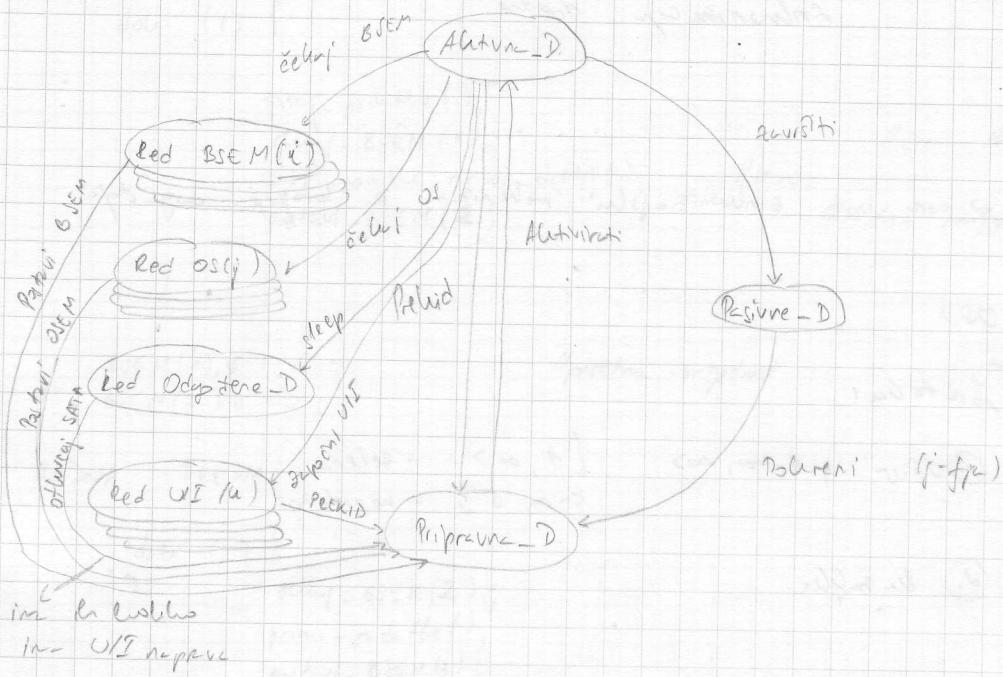
① aktivna

② priprema

③ postava

④ blokirana

- a) na OSEM
- b) na OSEM
- c) delu na budjenje (ekspresijem dretva)
- d) delu na survezu VI operacije



sleep = polusnit sebo

- Pripravne-D \Rightarrow red u vise procesora

- Aktivn-D \Rightarrow svaki procesor im svoj red (samo 1 D u redu)

Jedrine funkcije

• učinku u jedru zbiraju pod utjecajem preluda i tada se poziva jedrnu funkciju

f-fz

- polereni kontekst u opisniku Aktivn-D

(ukinje je te osi automatski
funkcije u programu)

- izlazak iz jedre ruk. te ne aktiviranje radne od skripti

\rightarrow a provodi aktivacije programe u redu pripravnih skripti, ali postupki
intervenije

Postupak aktiviranja:

- aktivirati prvu skriptu u redu Pripravne-D

1) prempost. 1. opisnik je Pripravne-D u red Aktivn-D

2) obnovi kontekst 17 opisne aktiva -D

3) vrati se u prelinatu detinu

- automatski projekti u mreži reda prelinute detinje
 - omogućiti poludanje alata po bilo kojim u prelinatoj detinji
 - aktivnosti odgovarajući (aktivni ili konzervativni) mreži reda
- OBNOVITI SADJAJ PC-a // tek na crizi

→ treba smet povezivanje /fiz

Ispitni - BJEM (I) ?

pohraniti kontekst u opisne aktiva -D;

alio je (BJEM [I], v = 1) ?

BJEM [I], v = 0;

obnoviti kontekst iz opisne aktiva -D;

y
vrati se u prelinatu detinu;

inče y

premetiti opisne reči aktiva -D u red BJEM [I]

aktivnosti prve detinje u red Prijavne -D;

y

Pojedinčni - BJEM (I) ?

pohraniti kontekst u opisne aktiva -D;

schedule ce odmeti

bježe detinje

ce se usmije

izvršenja (po prioritetu FIFO)

ili nešt drug)

premetiti opisne reči aktiva -D u red Prijavne -D;

alio je (red BJEM [I] neprazan) ?

premetiti prvi opisni reči aktiva -D u red Prijavne -D;

y inče y

BJEM [I], v = 1;

y

aktivnosti prve detinje u red Prijavne -D;

y

pharmaci kontakst u opisnik aktuacije
 $OS(j).v = OS(j)-1$
da je $OS(j).v >= 0$?

✓ /spiratice (77)

obavijeti kontakst u opisnik aktuacije

(a priori)

varijacije u prethodnoj detekciji

inicijal

prethodni opisnik je red aktuacije u red os(j);

aktivacija novih detektora na red prijavljene;

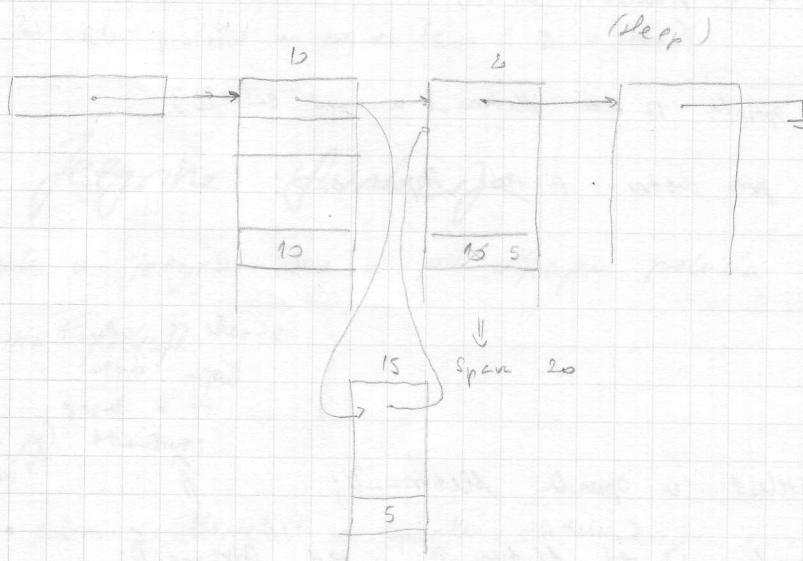
y



$$OS.v = -2$$

detekcija $-OS(j) \Rightarrow OS.v = -3$ zato je 0 (FB je 105 cm/s) nista može reći

Funkcije za otvaranje kamenje



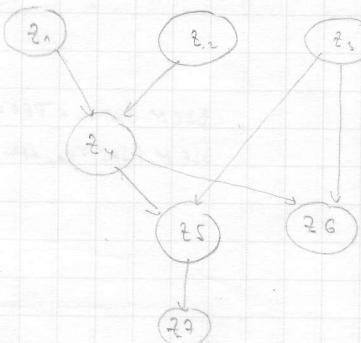
SEM	j-fis	ZELENS	CRUENO		
			NEIMA NIWOGA MA NEIWA U REOU	projekt u red BSEM;	
BSEM	čekaj-BSEM	v=0; projekt semfor;			
	postavl-BSEM	- (v=1);	v=1; preprekt 1. opisnik u BSEM u red Preparuna-D;		
OS	čekaj-OS	v--;	v--;	projekt u red OS;	
	postavl-OS	v++;	v++;	also je (v=0) ; preprekt 1. opisnik a -> Preparuna-D; ;	
OSEM	čekaj-OSEM	v--;		projekt u red OSEM;	
	postavl-OSEM	v++;	v++;	preprekt 1. opisnik OSEM -> Preparuna-D;	

Zadatak 5.1.

2. izvođenje zbiru dretki prema slti. (izgubje red 3.1.) konzult se
semfori.

- Kako je semfor po potrebi u funkciji?
- Poštne vrijednosti?
- Neka je T_1 tekuća potreba i. Projekti sviši zadatki T_1 u T_1'

3. minimalizacija broja novih funkcija semfora.



	BSEM	OS	OSEM
a)	T_1 (broj strujanja) (24-27)	4	4
b)	svi $v_0 = 0$; $v_1 = -1$; $v_2 = -1$ $v_3 = -1$; $v_4 = 0$	$v_1 = -1$; $v_2 = -1$ $v_3 = -1$; $v_4 = 0$	svi $v_0 = 0$;
c)	$T_1' = T_1$; $T_2' = T_2$; postavl-BSEM(2); postavl-OS(1);		
			-/-
		$T_4' = \text{čekaj-OS}(1)$;	$T_4' = \text{čekaj-OSEM}(1)$;
		$\text{čekaj-BSEM}(4)$;	$\text{čekaj-OSEM}(4)$;
		T_4' ; postavl-OS(2);	
		$\text{postavl-OS}(3)$;	
		$\text{postavl-BSEM}(6)$;	

Zadatak 5.2.

Problem pušenja cigareta

Zamislimo da su 3 dječaci pušaci i 1 dječak državačem. Smrši pušaci neprestano skupljaju cigarete i puši. Kako bi se smrši ogreba potrebno je imati:

3 dečka: duhan, papir i šibice. 1 pušac može u nezgrijanju boljševicu

šibice, papir, državci, duhan, i to treći ima šibice. Državač može imati

šibice u nezgrijanju boljševicu. Tjedec nemire stavlja na stol

2 dečka dečaka. Posao kog je treći dečak signalizan državcu,

smrši ogreba i puši. Državač stavlja novi 2 dečka na stol "

abuši te ponuge. Na poslušnu stol je pušak. Napred dječaci

pušaci i državci dolaze da se one medusobno ispravljaju sigurnosti

o svog bitnog značaja.

Službeno: 1) redatelj radnje istupaju po opisu u teliku za dječake

2) medusobno se istupaju boriste zapovjednicu medušu

3) nemire biskvunčne petlje (rečenje)

4) nemire potresne žestige

5) nemire redog očkunje

dječak državač ()

while (1) {

čekaj - BSEM (STOL-PRATAN);

stavi 2 dečka na stol;

postavi - BSEM (SASTOJCI-NA-STOLU);

BSEM (STOL-PRATAN), p = 1;

BSEM (SASTOJCI-NA-STOLU), p = 0;

dječak pušac ()

dol (1) {

čekaj - BSEM (SASTOJCI-NA-STOLU);

abu su ponutili dobiti dečake;

postavi - BSEM (STOL-PRATAN); //dignuti rukoj državcu

smrši cigarete i puši; }

? } Inace { postavi - BSEM (SASTOJCI-NA-STOLU); } ??

- brod deluje (npr. ako je dan posl. i apot u njegovu testuju na stolu)

detekcija $\text{Rjaci-s-sibicane} \cap$

detekcija

čekajući - BSEM (DUVAN-PAPIR);

uzmi se stopne;

postavlji - BSEM (STOL-PRAZAN);

postavi

BSEM (STOL-PRAZAN), $v=1$;

BSEM (DUVAN-PAPIR), $v=0$

BSEM (DUVAN-SIBICE), $v=0$

BSEM (PAPIR-SIBICE), $v=0$

detekcija Tryvac

detekcija

čekajući - BSEM (STOL-PRAZAN);

stavi se stopne;

postavlji - BSEM(); \Rightarrow jedan od 3

1-7.2 Otkrivanje satova?

počinjati kontekst u opisniku aktivnosti;

Aktivnost. Otkrivanje --; $\rightarrow ?$

Vreme otkrivanja

also je (ted odgovoreno nepravilno) ?

Zadnje - klasnjenje --;

also je (rezultat ==) {

premetkuti opisnik iz Aktivnosti u Pripreme;

premetkuti 1. opisnik i sve sljedeće iz Odgođene;

algoritam za
 raspodjeljivanje poslova
 će održati bilo

u Pripreme tako da im je klasnjenje = 0;

premetkuti 1. opisnik iz Pripreme u Aktivnosti

}

also je (Aktivnosti se raspodjeli podjela vremena &8 otkrivanje)

Aktivnost. Otkrivanje == 0 } }

Aktivnost. Otkrivanje = MAX - dobitak - za - tu - detektuju;

premetkuti opisnik iz Aktivnosti u Pripreme;

premetkuti 1. opisnik iz Pripreme u Aktivnosti;

}

obnoviti kontekst iz opisnika aktivnosti;

vratiti se u prethodni detektuju;

}

- poseg je ve prethodno per procesor automatski zaboravlja prethide led te detektuju.

prethid

\rightarrow što da se učeti više procesorski?

\rightarrow steklenički putem mrežnog isključivanja

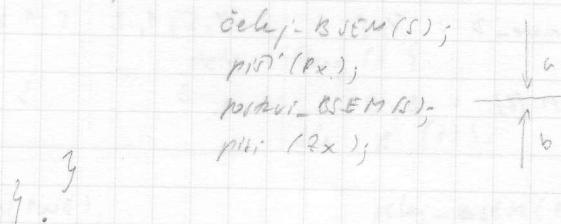
- variabilni OGLEDAD - JEZGRE - rezervacija biva na svim potku treba

Kriterij za klasificiranje Antrop (osim TAS, i sljedećih antropozna)
+ redovne delovanje

Zadatak 5.3.

a) Tri procedure pozvane poredjajuju se u tri skupine D1, D2, D3, s
prirodnim redoslijedom 1, 2 i 3. Koji je prirodnji red? Svi zadataci biju obliku
dove u istog obliku $D_x \rightarrow y$

Dove je (1) ?



Istodobno se poziva i red semafra u prirodnim poslovima je dosta
one luge je prav u redu pozivanih (nema posebnog reda likovnih dretva).

Prije poštovanja autora dretva, semafra S je reprezentovan ($v=0$), kolonu kolonu
vremena one drette se nachaju u redu semafra S. Kao se tada pozove procedure
(f-fn) potiski-BSEM(S), išta će se ispisati u rezultatu?

Red	BSEM(S)	BSEM(S)	Tripravne-D	ispis	
				P	Q
3. 2 1	0	0	1 -		
2 1	0	3a		P3	
1	0	3b 2a			23
3 1	1	2a		P2	
1	0	3a 2b		P3	
-	0	3b 2b 1a			23
3	0	2b 1a			22
3 2	0	1a		P1	
2	0	3a 1b		P3	
-	0	3b 2a 1b			23
3	0	2a 1b		P2	
2	0	3c 2b 1b		P3	
-	1	3b 2b 1b			23

Novi red do beskonačnosti

ispis

→ M1-in (je 4. red 5. dretvi)

→ 1. se treba pozvati od 1-4

Zadatak 5.4.

U prezentovanom trendu stanja skupine je sljedeće: D1 iž adfina, D2, D3 + u redi OBSEM(1) + D4, D5 u redi
Pripreme skupine. Pošto redi adfina skupine 1 posune prava.

postaviti OBSEM();, čime će mogućnost struktura postolja progne
nizom programne funkcije (KIFD, svr, rekvizit).

Prije poziv Postovi-BSEM(1):

Postoli:

$$\text{Adfina-D} = 1 \quad - \quad 4 \quad | \quad 4$$

$$\text{Red Pripreme-D: } 4,5 \quad 4,5,1,2 \quad 5,1,2 \quad | \quad 5,1,2$$

$$\text{BSEM(1): } 2,3 \quad 3 \quad 3 \quad | \quad 3$$

Zadatak 5.5.

Modelirati vrtuljku s 2 tipa skupina: skupina Projekti te skupina Vrtuljku. Skupine Projekti su nekoje skupovki ulozi na vrtuljiči kde vise nemoj prenosi (koristi se poslovni ulogibar organizacija vrtuljku polaznik) ali kada je pun. Za skupovke ulozi koriste se nekoje i oblike varijable.

Deklaracija Projekti()

```
    Obezj-OSEM(vrtuljku);
    sredni();
    Postovi-OSEM(sred);
    Obezj-OSEM(knj);
    ustani();
    Postovi-OSEM(sifra);
```

}

postovi
vrijednosti:

Deklaracija Vrtuljku()

```
dok je (1) {
    for (i=1 do brojesta) postavi-OSEM
        (vrtuljku);
    for (i=1 do brojesta) oduzj-OSEM(jed);
    polazni-vrtuljku();
    zavrstvi-vrtuljku();
    for (i=1 do brojesta) postavis-OSEM (korj);
    for (i=1 do brojesta) deluj-OSEM(sifra);
}
```

OSEM(vrtuljku).j=0;
OSEM(jed).j=0;
OSEM(korj).j=0;
OSEM(sifra).j=0;

6. Međutvrzna komunikacija ;

konceptualna monitora

MS [] → međupremnica ("oo" mjesto)

zadacha 8.3. (a)

U sistemu postoji 2 vrste deictivi. Ping (ispiraju "Ping") i Bang.

Imenjivosti ovete plus da one ispiraju Ping - Bang, ... itd.

deictiv Ping () {

dok p (1) {

otkup - BSEM (PONG);
ispisi ("Ping");
postavi - BSEM (PONG);

y
y

deictiv Bang () {

dok e (1) {

otkup - BSEM (PING);
ispisi ("Bang");
postavi - BSEM (PONG);

y
y

8.V.:

BSEM (PONG), r=1;

BSEM (PING), r=0 ;

• "Ping Ping Bang" (za dr.)

deictiv Ping () {

dok p (1) {

otkup - OSEM (1);
ispisi ("Ping");
postavi - OSEM (2);

y
y

OSEM (1), r=2;
OSEM (2), r=3;

deictiv Bang () {

dok e (1) {

otkup - OSEM (2);
ispisi ("Bang");
postavi - OSEM (1);
postavi - OSEM (1);

y
y

Pravotnoč (1) }

dale je (1) s

postavi pun (P);

čekaj - OSEM (PUN);

čekaj - BSEM (1);

$$MS(U) = P; \\ U = (U+1)/n;$$

} k.o.

per im
više postavila

Postavi - BSEM (1);

Postavi - OSEM (PRAZAN);

y

Potpore (1) }

dale je (1) s

čekaj - OSEM (PRAZAN);

čekaj - BSEM (2);

$E = MS(1)$;

$$17 = (17+1)/6 n;$$

? k.o.

Postavi - BSEM (2); } postavi OSEM (PUN);
postavi - pun (e);

y

• OSEM (PRAZAN), $r = 0$;

OSEM (PUN), $r = n$;

→ više postavila t.i. potpora \Rightarrow potreban je 1 BSEM za pravotnoč i potporu

Postavi testop

memorijska celica A

MA

BSEM(A), $r = 1$;

MB

BSEM(B), $r = 1$;

Pray 1.:

čekaj - BSEM(A);

čekaj - BSEM(B);

... → Memorisi MA i MB

Pray 2.:

čekaj - BSEM(B);

čekaj - BSEM(A);

\Rightarrow potpni testop

... → Memorisi MA i MB;

postavi - BSEM(A);

postavi - BSEM(B);

postavi - BSEM(A);

postavi - BSEM(B);

zadatci 6.5.

finiširajući deute vode i kisika spoje stvari molekule vode.

dretci H(1)9

Poštovi - OSEM (stigao H);
čekaj - OSEM (spajanje);
postoji se s atomom O(1);

deute O(1)9

čekaj - BSEM (K.O.);
čekaj - OSEM (stigao H);
čekaj - OSEM (stigao H);
Poštovi - OSEM (spajanje);
Poštovi - OSEM (spajanje);

y
P.V.

OSEM (stigao H), $r=0$;
OSEM (spajanje), $r=0$;
BSEM (K.O.) $r=1$;

V spoji atore u $H_2O(1)$;

poštovi - BSEM (K.O.);

)
sprječava da se vide
dretci kisika približavati

H:

čekaj - OSEM (spajanje);

postovi - de 1;

poštovi - OSEM (stigao H);

O:

čekaj - BSEM (K.O.);

Poštovi - OSEM (spajanje);

Poštovi - OSEM (spajanje);

čekaj - OSEM (stigao H);

čekaj - OSEM (stigao H);

spoji u H_2O ;

poštovi - BSEM (K.O.);

Nučni ujetci a nastavni potporuč začeti

- ① u nekom trenutku nema sredstva sa mračnimi medusama i iglavama
- ② dretci + sredstva ne nose odvrat.
- ③ dretci drži dolijevi pri sredstvu dok deba se dođelu drugih sredstava

Monitors

- program mechanism to synchronization objects

- Hoare, 20-16

- describes how shared resources are protected by monitors

defining $f_{lock}(i)$

do \downarrow (i) {

 miss(i); // N.N.D.

 | m-hr. $U_{seti_stropic}(i)$;
 | $H_{st}(i)$; // R.O.

 | m-fre. $S_{seti_stropic}(i)$;

y
y

1

* m-hr = monitoring function \rightarrow changing resource state P.P.P

↳ works: f-free

Structure protocols:

(1) Monitor's structure (from binance)

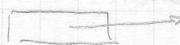
Monitor [M]



$v = 0 \text{ ili } 1$

(2) Red update

Red_update [M, 1]



Red_update [M, 2]



\Rightarrow vice slovenski update

für die unterschieden Monitore

variable Schleifenanzahl

- Uti-n-monitor (M); \rightarrow mutex-lock ($\&M$);
- Iaci-n-monitor (M); \rightarrow mutex-unlock ($\&M$);

identische Funktion
ablag-SSEM

\rightarrow rechte Maus, drap rechte Maus, bewege

strukturelle parallele Monitore (hier zwei ablag, ein weiterer SMM)

- Vrstti-n-red-wyeta (M, K);

↓
richtig los, red
wyeta

- Čekti-n-red-wyeta (M, K); \rightarrow cond-wait ($\&M, \&K$);

↳ Wichtig! nur funktioniert vrssti-n-red-wyeta

↑
up zu einem Punkt in Ausführung werden

- Oslobodi-n-red-wyeta (K); \rightarrow cond-signal ($\&K$);

- Oslobodi-n-sve-n-red-wyeta (K); \rightarrow cond-broadcast ($\&K$);

für Čekti-n-red-wyeta:? (nur marginale Funktion) → Cerniček nicht mehr da (?)

Vrstti-n-red-wyeta (M, K);

Uti-n-monitor (M);

Blau P monitor sorgt für dringende Aktionen
rote u. monitor

m-fje \rightarrow monitorische Funktionen folgen so a transmission norms
rada) može biti prekrnute

• PRAVILA koja TREBA POSTOVATI PRILIKOM IZGRADNJE MONITORA:

- 1) monitorische Funktionen je zadidena monitorom sa formom
- 2) monitorische funkcije prave limitiranu objektu raspodelu ujet;
- 3) monitorische funkcije prave limitiranu objektu po potrebi odrzavaju
detake iz redov ujet

drevni filozof (i) {

dole (1) {

mislišti();

m_uzeti_stapice();
jesti(); \neq k.o.

m_spusti_stapice();

4) raspodeljuje ujete je oblik je "dole je",
a ne je "nemo je"

\rightarrow drevni filozof im suvi
red ujet?

m_uzeti_stapice () {

mutex_lock(&k);

while (biti_ljepi (od_du) štapić_uzeti) {

cond_wait(&red[i], &k);

uzeti_stapice();

mutex_unlock(&k);

m_spusti_stapice () {

mutex_lock(&k);

spusti_stapice();

cond_signal(&red[(i-1)%n]); }]

cond_signal(&red[(i+1)%n]); }] održavanje
sezida

mutex_unlock(&k);

to je trebati if-ovi jer
se može ovde detektirati ne trebati
postati (ali se im nisu dostupna
obi stvari)

6.1

Problem shry mstz

flori' most fi whi mstz . Na nenu ntnemene smpr btsi nprse 3
 automobile busi note a other smjeru dinlunashd' automobile,
 dntre Auto / zedns. Hypoth' prendlwue fra soper-nc-mstz
 i' pslis-mstz.

Dntre Auto (smper) { // smper oltz }

m-pplni-st-nmstz (smper);

ppln-mstz; MKJ

{ } = m-fidi-n-mstz () ;

}

m-pplni-st-nmstz (smper) {

mutex-lock (&k);

dk (anti == 3

|| (smper-nc-mstz = smper 88 anti > 0))

cond-wait (&red, &k);

anti++; smper-nc-mstz = smper;

Oreni-prela mstz;

mutex-unlock (&k);

y

P.V.;

anti = 0;

smper-nc-mstz = 0;

k

red-uveta

red

→ variable zuljevarsa

mutex-lock (&k); } if (anti == 0) smper-nc-mstz = smper;

anti--;

cond-broadcast (&red); } No kibodi ne

mutex-unlock (&k);

obs bi
je moglo

start de N

jude do

starvata nt

Zadatak 5.2. (MS i Linux programeri)

U idući zadaci može MS i Linux programeri, izgraditi ima 1 performan logički ciklus. U tom ciklusu u redovanu smjer biti: krov 1 vrati programeru ili je redovan prazan.

Programer (vrati) :

```
vrsta = 0 //MS , 1 Linux  
vrati (vrsta);  
obavij; }  
izdati (vrsta);
```

vrsta = 0 //MS , 1 Linux

M -> var zadržavanje

uv[2] -> 2. reda upita

int čekac[2] = {0};
int br[2] < 80;
int siti[2] = {0};

a) synchronizacija programera monitorom

```
m-udi (vrsta);  
mutex-lock(&M);  
čekac[vrsta]++;  
dak je (br[vrsta+1] % 2) > 0
```

m-izdati (vrsta);
mutex-lock(&M);

br[vrsta]--;

if ((siti == n) && (čekac[1-vrsta] > 0)) alio je (br[vrsta] == 0) {

siti = 0;

cond-wait (&uv[vrsta], &M);

cond-broadcast (&uv[1-vrsta], &M);

```
br[vrsta]++; čekac[vrsta]--;  
dak je (čekac[1-vrsta] > 0) siti++;  
mutex-unlock(&M);
```

mutex-unlock(&M);

3

b) raditi zadataku bez pojave izjednačavanja, tako da napise u programera iste vrste pede ipa ond idu drugii (ako deluju u redni)

programera iste vrste pede ipa ond idu drugii (ako deluju u redni)

③ U svrhu potjeće da vole djetki ping i pong. Daje ih u spremu

- funkcionizirati monitorima kuo da će ispisati „Ping Pong Ping...“

Ping() {

white(1) {

mutex-lock(&M);

du je (ZADNE = PING)

cond-wait (&Hd[PING], &M);

- //

ispis;

ZADNE = PING;

cond-signal (&Hd[PONG], &M);

mutex-unlock(&M);

} int ZADNE = PONG;

#define PING 1

M

Hd [2]

#define PONG 1

8. Gorodenje spremidium program

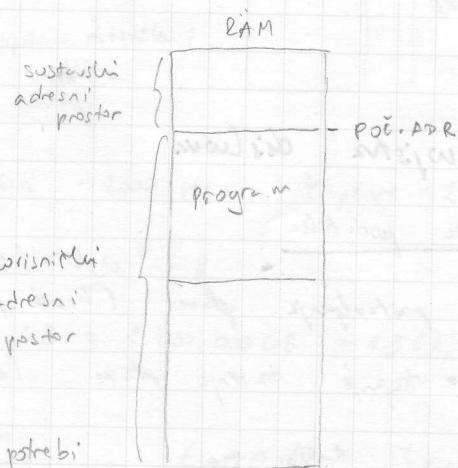
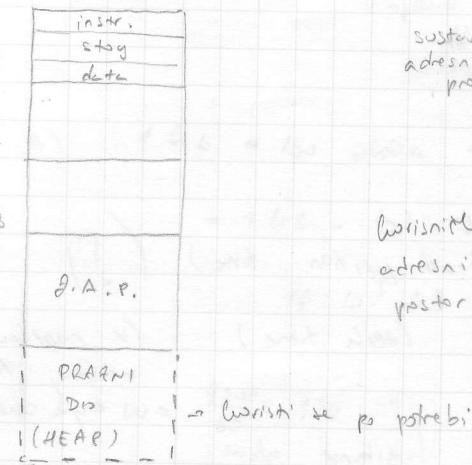
• Adrese se generiraju unutar procesora:

1) PC

2) SP

3) adrede generiranje instrukcijama

procesorski adr. prostor:



ovduće je bilo napisati → oskoli programi koji deluju na izvodenje u procesoru spremidium

↳ programi nisu mogli biti veci od rezervacije prostora u RAM-u

Obrav svjete magnetične diskove

članak \Rightarrow slup staze isti poluprera

sektor \Rightarrow dio staze

\downarrow

unutri imaju kapacitet (oko 1GB) (1-4 GB)

• cluster \Rightarrow nizova sektora (1, 2, 4, 8... sektora)

reper \Rightarrow najviši početak (prije sektora na stazi)

Afređivanje sektora

1.) RBR place

2.) RBR staze

3.) RBR sektora na stazi

- tipični promjer plitki: 2,5, 3,5 inč

- gustoća staze 1000tbi obo 250 000 fitaza po inču radijus (2.5 TB disk)

- gustoća bitova na stazi je 10-500 gigabit po inču²

- disk ima 1-20 plitki

- staze ima 120-300 sektora

- brzina okretanja u min: 5400, 7200, 10k

\rightarrow sektor veličine 512B 16 4KB

Vremenske svjiste diskova

Ukupno trajanje prijenosa podataka

■ trajanje postavljanja glave (head position time)

• trajanje travljenje staze (seek time)

(u pogledu staze

u definisano vremenu od
broja okupljenih stazi)

1. ubravanje

2. konstruktivna

3. uspon

4. fin raspoređivanje

gibanje glave

• rotacioni čas vrijeme T_e , ($T_e = \frac{T_d}{2}$) (rotation

latency)

■ trajanje prenosu podataka (Data Transfer Time)

- trajanje otvaraćeg ciklusa stanja
- trajanje prenosnog glave na stanje na stanju + $\frac{T_B}{2}$

①

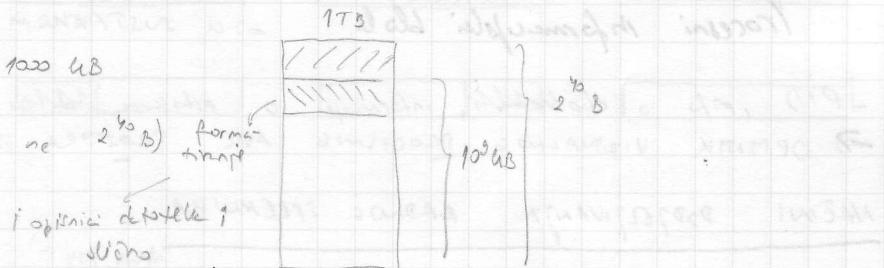
Daje se da ima 100 sektora po trakama volumine 1 GB, 2000 stanja po placi, 4 place i vrati se brzina 2200 obratova u min. Podeliti se papirski u obje stane place. Upravljanje diskop je potreban 1 ciklus stanja u intervalu spremanje, a zatim je potreban u gornji spremanje. Papir je gornji spremanje odvaja se brzina od 20 MB u sekundi. Ako je vrijeme ulaza u mreže isto i izlaza.

a) Koliki je kapacitet disk?

b) Koliki je vremenski razmak premeđujući kompletne smjestitele do sredine volumene 135 GB, u kome je trajanje prenosnog glave stanja na stanju 1 ms.

10ms i vrijeme prenosnog glave se stanje na stanju 1 ms.

$$10^3 \cdot 10^6 \cdot 10^{10} \text{ B} = 10^{19} \text{ B}$$



a) $1 \text{ GB} = 100 \text{ sektori} \cdot 2000 \text{ stanja} \cdot 4 \text{ place} \cdot 2 \text{ stanje}$

$= 1 \text{ GB} \cdot 10^2 \cdot 2 \cdot 10^3 \cdot 8$

(c) $16 \cdot 10^5 \text{ GB} = 1600000 \text{ GB} = 1562,5 \text{ MB} = \underline{\underline{1,5625 \text{ GB}}} \quad \text{S.P.}$

b) $w = \frac{2200}{60} = 120 \text{ s}^{-1}$ $t_{\text{stan}} = 10 \text{ ms}$ $t_{\text{prem}} = 1 \text{ ms}$

$v = 20 \text{ MB/s}$

1) trajanje postavljanja glave: $t_{\text{prem}} + \frac{T_B}{2w} = 10 \text{ ms} + \frac{10}{2 \cdot 120} = 10 + \frac{10}{240} = 10,12 \text{ ms} \quad \text{P.}$

2) trajanje otvarača 

$T_L = 8,33 \text{ ms}$

\hookrightarrow trajanje 1 stanje

3) paralelna se odvija:

a) prijenos podataka

$$v = 20 \text{ MB} \quad \text{staza} = 100 \text{ UB} = 102400 \text{ B} = 102400 \cdot 8 \text{ biti}$$

$$t_p = \frac{s}{v} = \frac{102400 \cdot 8}{2 \cdot 10^7} = 40,96 \text{ ms/}$$

b) projekt glave te staze na stazu $t_{prem} = 1 \text{ ms}$

$$t_p > t_{prem} \Rightarrow t_p \text{ je užem u } \underline{0,052 \text{ ms}}$$

4) rotacioni učinkovi $T_R = 4,12 \text{ ms}$

5) otvaranje 2 staze (otvara se i zatvara)

$$t = T_R = 8,33 \text{ s}$$

$$6) t_p = \frac{35 \text{ UB}}{v} = \frac{35 \cdot 1024 \cdot 8}{2 \cdot 10^7} = 14,32 \text{ s}$$

$$\text{tukupno} = 14,12 \text{ s} + 8,33 + 40,96 + 4,12 + 8,33 + 14,32 \text{ ms}$$

$$\text{tukupno} = 28,54 + 16,66 + 45,13 = 45,20 + 45,13 = 90,33 \text{ ms}$$

Procesni informacijski blok \rightarrow u sustavu ADR. procesora

- PID, IMP o adresama, informacije o adresama učite procesor učitati
 \Rightarrow OPISNIK VIRTUALNOG PROCESINGA ADR. procesora

NACIONI DODJEGLJIVANJA RADNOG SPREMIKA

- svaki program ima relativne adrese (počinje od nule \Rightarrow PA je apsolutna adresa)

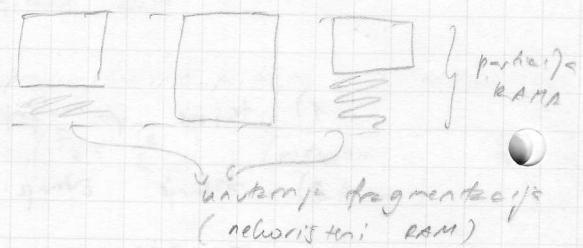
- za svaku adresu u programu mora postojati moćna logika koja će je prebaciti u adresu u apsolutnom ili relativnom obliku

- prije poštovanje se svakoj relativnoj adresi mora dodati PA

Statičko raspoređenje:

2 vrste fragmentacija:

1) unutarne \Rightarrow



2) varijacije \Rightarrow programi se raspodjeljuju po raspodjelama

\rightarrow auto radi program iste u redi, negativno

raspodjelje se ostatak prazne (nisi du RAMA)

Dinamičkih upravljanja \Rightarrow proces generira logičku adresu i "true, false"

ne je definiran se postavi stvarajući logičku adresu u CA

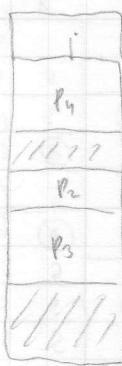
če PA je rezervirana procesoru ($PA \rightarrow$ bazu! register)

\rightarrow proces ostaje u logičkoj adresi . * menjati se prilikom raspoređivanja

\rightarrow RAM se ne mijeni u raspoređuju

→ ali se menjaju direktni istog

prosesa indeks se ne menjava



\rightarrow fragmentacija, krovni: 50% prazno \Rightarrow broj rupa jednako je 50%

Broj praznih blokova

$\rightarrow \frac{1}{3}$ svih blokova su rupi



oslobodjenje bloka



$$m = a + b + c + d$$

\rightarrow prije rupi

$$n = (a + c + 2d) \cdot \frac{1}{2}$$

\rightarrow rupa

$$b = c$$

$$m = a + 2b + d$$

$$n = b + d$$

P_1 - vjerojatnost zahvata u spremniku

$$P_0 = -1t \text{ oslobodjanje bloka}$$

$$P_2 = P_0 \Rightarrow$$
 u svakim statičkim rukovodstvima

$P_1 \rightarrow$ vjerojatnost da će postići broj rupa

$$P_+ = P_0 \cdot \frac{a}{m}$$

$$P_- = P_0 \cdot \frac{d}{m} + 0$$

Vjerojatnost da je proces rupi (oni koji zahvatuju spremnike)

$$P_+ = P_-$$

$$P_0 \cdot \frac{a}{m} = P_0 \cdot \frac{d}{m} \Rightarrow \underline{\underline{a=d}}$$

$$\underline{\underline{m=2b+2d}} \quad \underline{\underline{n=b+d}}$$

$$\underline{\underline{(m=2n)}}$$

8.2

Adušen je od stranice, program učita se u mem' (1-400) generira

sljed. adrese: 23 47 333 81 105 1 } 400 157 30 209
 (1) (1) (1) (2) (3) (1) (8) (4) (1) (5)
 289 145 360
 (6) (3) (6)

Adušen generira rabiću je stranicu. Program ima ne raspodjeljuju

400 rješ' primarnog spremnika. Njihovih niz referenciranih stranica

velike su rješi. Moliti je potreba pravljaz za sve strategije

Abrage se stranice.

adresa	23	47	333	81	105	1	400	157	30	209	289	145	360
zadržev. u stranicu	1	1	2	2	3	1	8	4	1	5	6	3	2
FIFO	-	1	-	1	1	-	(8)	8	8	8	(6)	6	6
-	-	-	2	2	2	-	7	(4)	4	4	(3)	3	-
-	-	-	-	3	-	-	2	2	(1)	1	1	1	(8)
-	-	-	-	-	3	-	3	3	3	(5)	5	5	5
LRU	-	1	-	1	1	-	-	1	1	-	1	1	(3)
(Least recently used)	-	-	2	2	2	-	(8)	8	-	8	(6)	6	6
-	-	-	-	3	-	-	2	(4)	-	4	4	(3)	3
-	-	-	-	-	3	-	3	3	-	(5)	5	5	5
OPT	-	1	-	1	1	-	-	1	1	-	1	1	-
(Optimalne)	-	-	2	2	2	-	7	(9)	-	(5)	(6)	-	-
de očekuje se retele	-	-	-	-	3	-	(8)	8	-	8	8	-	-
-	-	-	-	-	-	-	3	3	-	3	3	-	-
LFU	-	1	-	1	1	-	-	1	1	-	1	1	1
(Least frequently used)	-	-	2	2	2	-	(8)	8	-	8	(6)	6	6
-	-	-	-	3	-	-	2	(9)	-	4	4	(3)	3
-	-	-	-	-	3	-	3	3	-	(5)	5	5	(8)

↑
Abrage se
boljši (naj referencirani)

13 zadržev.

① Prioritete stranici: zapis tablice presedenca programa na cravi primjene optimalne strategije.

200 stranice (adresa)	oluvir	bit prioritetnosti
1	1	1
2	3	0
3	4	1
4	2	0
5	2	0
6	2	1
7	2	0
8	3	1

Fakti mehanizam

→ funkcija CRU algoritma

→ sve obranice se menjaju u crnu CRU

→ gled se samo A - register, (access, ali se koristi u problemu kočenja)

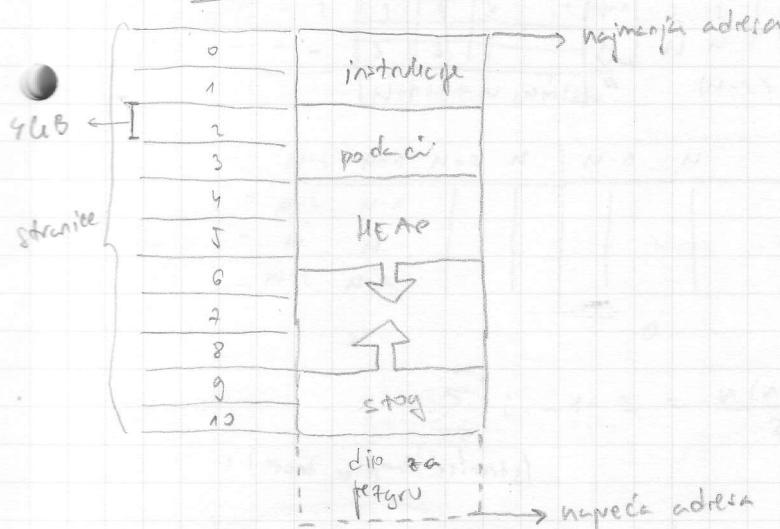
čuo je $A=1$, A je redire

ak nebiti dolje od beskonačne
putanje (koju su se sve stranice može u problem stvaraju)

→ nivo de one obranice
koristi u \wedge otkup
tako

putanje (koju su se sve stranice može u problem stvaraju)

Proces



	RAM
0	00
1	00
2	00
3	.
4	1
5	5
6	4
7	0

3 4 kB

oljni

Tablica prestonje (svaki proces je treba imati) → broj redaka jednako
broju stranica

BP	7	1
0	7	1
1	4	1
2	X	0
3	X	0
4	6	1
5	5	1
6	X	0
7	X	0
8	✓	0
9	✓	0
10	X	0

X - nebita adresa (nebitno)

→ TABLICE PRESTONJE

⇒ polazne u stranice logičke adre
prošire operativne sisteme

CR3 → vratačka na tablicu prestonje
Lokacija je u procesoru

svaki dñ

→ veličina stranice ⇒ potencija broja 2

→ tablica prestonja (a kojem je oljni logik stranica) ⇒ smjesta u spremništu
neovlašćuju

↳ ima u sebi BIT PRISTUPA

8.3

U sustavu s virtuelnim spremnikom veličina olvira je $N^2 \cdot N^2$, a
olviri se bude na zaliđev. Algoritam tempe se odvijače je CRL.

Predak A[1...N, 1...N] je pohranjen pod redicima,
u uspešnici kolica je mjeri broj redic u matrici. Koliko parametara
je potreban redic program koji će predati t o redom
spremnika postoji:

a) jedan i olvir

b) 2 olvira

c) 3 olvira

d) k olvira

	1	2	3	\dots	$N-1$	N
1	1	0	1	0	0	0
2		1	0	0	0	0
3			1	0	0	0
\vdots				\ddots		
$N-1$					1	0
N						1

$$t = 0$$

$$2a \quad i = 1 \text{ do } N-1 \}$$

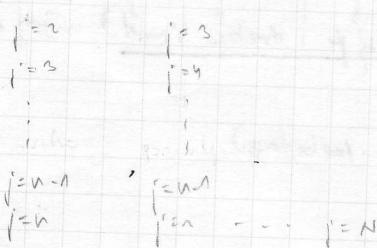
$$2b \quad j = i+1 \text{ do } N \}$$

$$t += A[i][j]$$

$$t *= A[j][i]$$



$$a) \quad i=1 \quad i=2 \quad \dots \quad i=N-1$$



(stranice koje se traže):

$$\underline{\underline{1 \ 2 \ 1 \ 3 \ 1 \ 4 \ \dots \ 1 \ N}} \quad i=1$$

$$\underline{\underline{2 \ 3 \ 2 \ 4 \ \dots \ 2 \ N}} \quad i=2$$

$$\underline{\underline{(N-3)(N-2)(N-3)(N-1)(N-3)N}} \quad i=N-3$$

$$\underline{\underline{(N-2)(N-1)(N-2)(N)}} \quad i=N-2$$

$$\underline{\underline{(N-1)(N)}} \quad i=N-1$$

$$S = 2 \cdot (N-1) + 2 \cdot (N-2) + 2 \cdot (N-3) + \dots + 2 \cdot 3 + 2 \cdot 2 + 2 \cdot 1$$

$$S = 2 \cdot [1 + 2 + 3 + \dots + (N-2) + (N-1)]$$

$$S = 2 \cdot \frac{n \cdot (n-1)}{2} = \underline{\underline{N(N-1)}}$$

$$b) \quad \begin{array}{c|ccccc|ccccc|c} & 1 & 2 & 3 & \dots & N & | & 2 & 3 & 2 & 9 & \dots & 2N \\ \hline - & 1 & 1 & | & 1 & & | & 2 & 2 & | & 2 & & | & 2 \\ - & - & 2 & | & 3 & \dots & N & | & N & 3 & | & N & & | & N \\ \hline & N & \text{proposition} & & & & | & N & N & & & & & | & N \end{array} \quad \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \quad \begin{array}{c} N-3 & N-2 & N-3 & N-1 & N-3 & N \\ | & | & | & | & | & | \\ N-3 & N-3 & | & N-3 & | & N-3 \\ N & N-2 & | & N-1 & | & N \\ & & & N & & N \end{array} \quad \boxed{4 = (3-1)}$$

$$\begin{array}{c|ccccc|ccccc|c} & N-2 & N-1 & N-2 & N & | & N-1 & N \\ \hline N-2 & N-2 & | & N-2 & N-1 & | & & & & & | \\ N & N-1 & | & N & N & | & & & & & | \\ & & & & & & & & & & | \end{array} \quad \begin{array}{c} 3 = (2-1) \\ 1 \end{array}$$

$$\sqrt{\sum_{i=1}^N i - 2} = \frac{N(N+1)}{2} - 2$$

$$c) \quad \begin{array}{c|ccccc|ccccc|c} & 1 & 2 & 3 & 1 & 4 & 15 & \dots & 1 & N & | & 2 & 3 & 2 & 4 & \dots & 2N \\ \hline - & 1 & 1 & | & 1 & | & 1 & | & 1 & | & | & 1 & 3 & | & N-1 & 3 & | & N-3 & N-2 & N-3 & N-1 & N-3 & N \\ - & - & 2 & | & 2 & | & 4 & | & 4 & | & | & 2 & 2 & | & 2 & 2 & | & 2 & 2 & N-4 & N-3 & N-3 & N-2 & N-1 & N-3 \\ - & - & 3 & | & 3 & | & 5 & | & 5 & | & | & 1 & 1 & | & 1 & 1 & | & 1 & 1 & N & N & N & N & N-1 & N-1 \end{array} \quad \begin{array}{c} N-1+1=N \text{ proposition} \\ (N-2)+1=N-1 \end{array} \quad \boxed{4 = (3-1)}$$

$$\begin{array}{c|ccccc|ccccc|c} & N-2 & N-1 & N-2 & N & | & N-1 & N \\ \hline - & N-3 & N-1 & | & | & | & | & | \\ - & N & N & | & | & | & | & | \\ - & N-2 & N-2 & | & | & | & | & | \end{array}$$

$$2 \quad \quad \quad 0$$

$$\sum_{i=1}^N i - 1 - 3 = \frac{N(N+1)}{2} - 4$$

$$d) \quad \begin{array}{c|ccccc|ccccc|c} & 1 & 2 & 1 & 3 & 14 & \dots & 1 & N & | & 2 & 3 & 2 & 4 & \dots & 2N \\ \hline - & - & - & | & - & - & | & - & - & | & - & - & | & - & - & | & - & - & | & - & - & | & - & - \end{array} \quad \begin{array}{c} N-3 & N-2 & N-3 & N-1 & N-3 & N \\ | & | & | & | & | & | \\ N-3 & N-3 & | & N-3 & | & N-3 \\ N & N-2 & | & N-1 & | & N \\ & & & N & & N \end{array}$$

$\Rightarrow N \text{ proposition}$

8.4

U nekom sustavu treba je obaviti 4 programe P_1, P_2, P_3, P_4 , koji su u redoslijedu S_1, S_2, S_3, S_4 i u veličini od 10 MB. Odgovarajući polazak je: $t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8$. Programi se mogu izvršavati u skladu sa sledećim rasporedom:

t_1	P_1	polazet
t_2	P_2	polazet
t_3		zavrsava
t_4	P_3	polazet
t_5		zavrsava
t_6	P_4	polazet
t_7		zavrsava
t_8		zavrsava

(premalo) rezerviranje za korisničke

programe, Polazak: otvaranje RAM-a ako

a) slatko upravljanje velikom
seg. od 10 MB

a)

	10 MB + 10 MB		
t_0			
t_1	P_1		
t_2	P_1	P_2	
t_3		P_2	
t_4	P_3	P_2	
t_5		P_2	
t_6	P_4	P_2	
t_7	P_4		
t_8			

\rightarrow RAM (bez varijacije)

b) dinamičko upravljanje

	20 MB		
t_0	P_1		
t_1	P_1	P_2	
t_2		P_2	
t_3		P_2	
t_4	P_3	P_2	
t_5		P_2	
t_6		P_2	
	P_4 ne stani nigdje		

c) struktura (1 MB str.)

t_0	$P_1 P_1 P_1 P_1 P_1$
t_1	$~~~~~ P_2 P_2 P_2 P_2 P_2$
t_2	$~~~~~ - II -$
t_3	$~~~~~ - II - P_3 P_3 P_3$
t_4	$~~~~~ - II - ~~~~~ P_4 P_4 P_4$
t_5	$~~~~~ - II - ~~~~~ P_4$
t_6	$~~~~~ - II - ~~~~~ P_4$

\Rightarrow delje brisanje

t_7	$P_4 P_4 P_4 P_4 P_4 ~~~~~ P_4 P_4 P_4 P_4 P_4$
t_8	$~~~~~ ~~~~~ ~~~~~ P_4$

\Rightarrow nije obrisan

- ⑤ je reči očekivane sa stranice zadana po tabelice prezentacije
2. 1 proces, stampi očekivane očekivane, stampi rezultate, se stvara s
 3. izvršenje samim algoritmom. Kako će se očekivani menjati tako
 - proses novi zadani na instrukcije. Međutim, nema se nulti.
- stranice).

Tabelice prezentacija

0	0	2	1
1	1	0	1
2	2	8	1
3	2	6	1
4		0	
5	5	5	1

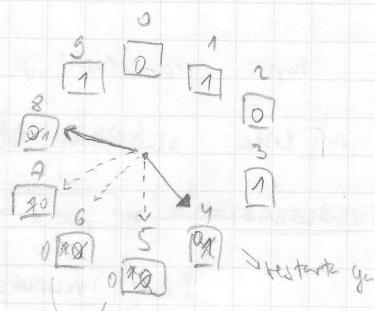
RAM

0	1
1	X
2	0
3	X
4	X
5	5
6	3
7	X
8	2
9	X

Niz instrukcija:

1. LDR R1,(508)
2. LDR R2,(332)
3. LDR R3,(256)

princip program
procesu



princip program
procesu
u memoriji (ne da ga nebi u memoriji
interpretuju da se
uvodi u 2-)

9. Datotečni podsustav

Opisnice datoteka

→ u datotečnoj tabelici

- naziv
- me klasična
- vrijeme zadnje promjene
- tip
- prvi pristup
- me zadnjeg korištenja
- lokacija
- vrijeme stvaranja
- i sljede (me ik gr)
- opis smjerstva (path)

ne svih su → datotečne tablice (Master File Table)

④ Unapređeni disk

- ② nečišća slobođenog prostora
- ③ opis slobođenog prostora
- ④ tablica opisnika

fp=fopen ("...")

↳ časiraj opisnicu datoteku u
datoteku → postava
puntor na svakom dijelu
opisnika gdje će nako
postaviti o datoteci (fseek)
ostalog za opisnik (lseek))

u tablici stranice \Rightarrow broj stranica (20 bita) + fastavice (12 bita)

1M stranica (4GB RAM, strana = 4KB)

\Rightarrow direktorijski adresari \Rightarrow 1024 blokova od 32 bita

Opis slobodnog prostora

\rightarrow kako je u detektaciji tablica (?)

= disk & sestopi od sektore / plitne
Linux Win

- ne podstavljači zapisi zwrotost: • BINARNI ZAPIS zwrotost (0 ili 1, bez zatvaranja)
- drugi način \Rightarrow lista slobodnih blokova



bit zwrotost: 0 1 2 3 4 5 6 7 8 ...
bit zwrotost: 1 1 1 0 0 1 0 0 0 ...

* Primer 9.1. (Unix det podatkov)

indeksni \in I-node \rightarrow indeksno dajuće blokove na tekture

sektor = 1K (1024 B)

32 bitna arhitektura \rightarrow velicina blokova 4B

\rightarrow u 1 sekutor stane 2^8 blokova

u I-node:

13 blokova \rightarrow 12 su prazne (prvi polozaj na sekutoru \rightarrow detektirati od 1K a mogu prazno zadefinirati)

11. \rightarrow polozajne - indeksne blokove

polozajne na sekutor 8 256 blokova koji polozajne na 1 sekutor - detektirati (256 u)

\rightarrow 11 blokova $= 256 \text{ u} + 12 \text{ u} = 268 \text{ u}$

12. \rightarrow doslovne - indeksne blokove

polozajne na sekutor 8 256 blokova koji polozajne na sekutore \rightarrow 256 blokova

Suma sljepce $\rightarrow 64 \text{ M} + 268 \text{ u}$ $\rightarrow 64 \text{ M}$

93. \rightarrow struktur - indirekt Datei

$$256 \cdot 256 \cdot 256 = (2^8)^3 = 2^{24} = 16 \text{ GB}$$

1 Sektor je 1 KB (zwei je 1 GB na Gruppe)

Speicher: 16 GB + 8 MB + 266 kB

\rightarrow -projektion unterteilt Fragmentar $\Rightarrow \frac{1K}{2}$. br - datei

Prinzip 9.6. NTFS

\rightarrow root \rightarrow Masterfile (nachspurktion: 1, 2, 4, 8, ...)

LCN \rightarrow Logical cluster number

datatabellen \Rightarrow MFT (Master File Table)

• opis slatcking partice i gdpje se detekcie vlastne

• opis u MFT je velikost 1 cluster (4 KB)

-also detekce sítne u -opis (u opisu), tvaru je smysl' (málo býť väčšie 4 KB za datatabu + 4 KB za opis)

Dekodovanie \rightarrow získanie podľa "virtuálnej" velikosti clusteru (VCN - Virtual cluster number)

HDD \rightarrow podelenie na clustre (such'šme svi LCN)

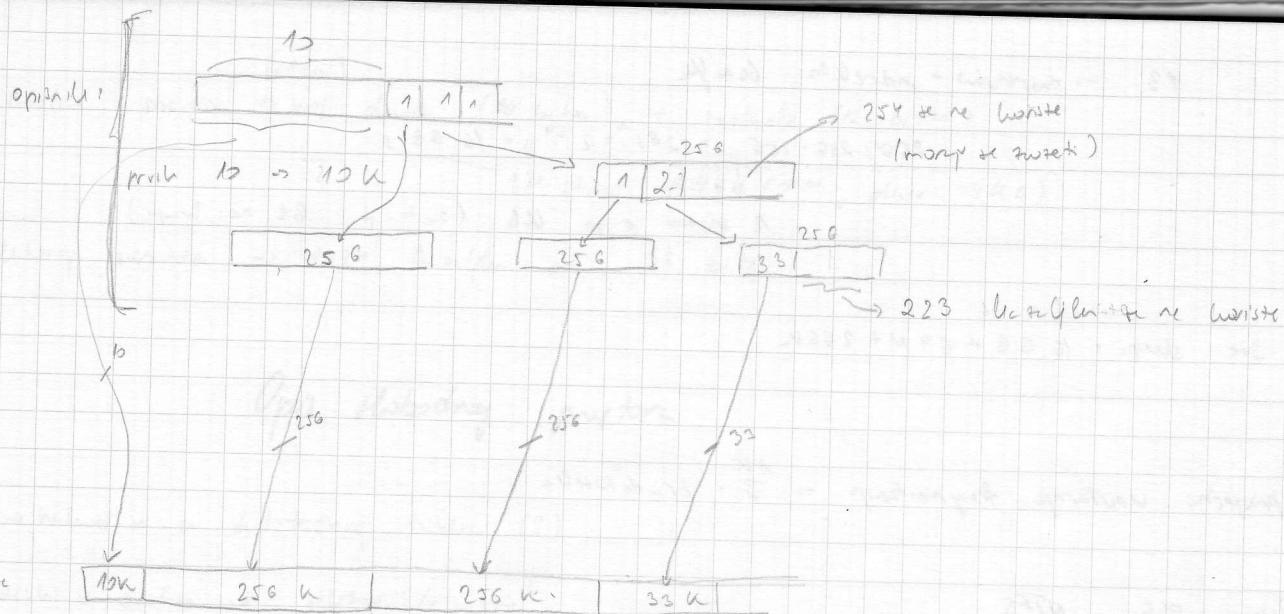
tablec.	VCN	Posledný LCN	Bry. slatck.
0	50		1
1	500		4
5	125		2

Prinzip 9.1.

Ako je v unix deklarovan podľa rozloženia oblasti v volume

555 KB, kde bolo spomínané posledné zadanie časť je dátoviu a adresáciu dátového systému. Služobnú organizáciu kódu.

Dátová línia	555 KB
Sektor	12248 (1 KB)
Kazeta	32 b (4 B)

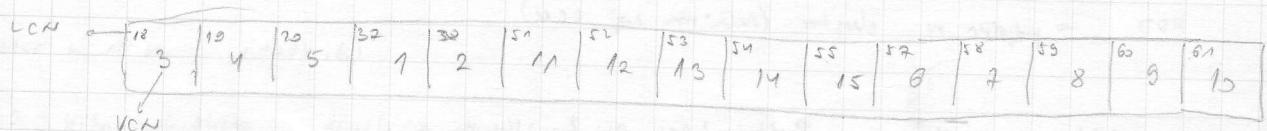


~ u opisivanju 13 blokova + 4 sektorja
po 4B

? minimalna te max. broj blokova: 13 blokova.

9.3

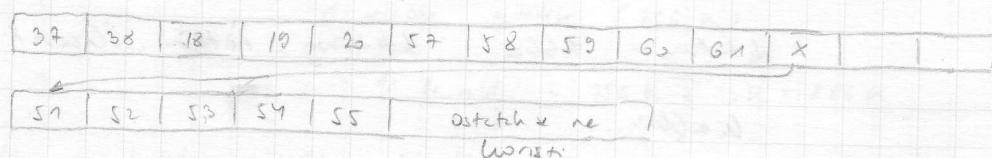
Nr. luka = clusters. Nakon pre sklapanja i slaganja u skupine i razdjeljivanje blokova mogu se raspodeliti u diskove. Optimalni skupovi blokova, tj. oni da koji imaju optimalnu raspodjelu. Clust 4K = 4K



NTFS!

VCN	LCN	Br.
1	32	2
3	18	3
6	52	5
11	1	5

UNIX: pravilni 13:



FAT - File Allocation Table

\downarrow $2^{20} = 1$ cluster + beschikbaar clusters (in opslagen op same��ke dan na 1. operat.)
 => max 4 kelen regelteile de totale

EOF → vrijbuit lege vrees funktie lege opeen is' de totale (niet beschik)

- fopen → API funktie

\downarrow
 VSOCL : file pointer => beschikbare opslink de totale

Zad 9.1.

Disk → platen van glas en hout (P=99). Nella de
 trekkers glas nummer 80, s. tim de høyeste nummer
 nummer 8. tallene på platen stårne: 3, 7, 45, 98, 71, 68, 20, 21, 5, 49.

Se svare og utleggje følgende iavutne:

a) uløyper når glas

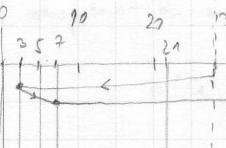
b) spesiell

c) slutt

d) Kva løype strategi mør det da regelkjører (stasjon)?

trekkers
platen

pos. nummer → 29



$$\rightarrow \text{totalt} \\ POMAN: 27 + 95 + 30 + 2 + 65 + 44 = 273$$

FCFS
(First come,
first serve)

→ normalt ja redusert
med nære doet da regelkjører

$$POMAN: 26 + 35 = 121$$

SSTF
(Shortest seek
time first)

→ nære doet da regelkjører

LOOK

$$POMAN: 68 + 35$$

kom u
sentrar u bogen i brenn
ide
do
redusert
teknikk

- SCAN
- ide do bane

$$POMAN: 70 + 95 = 165$$

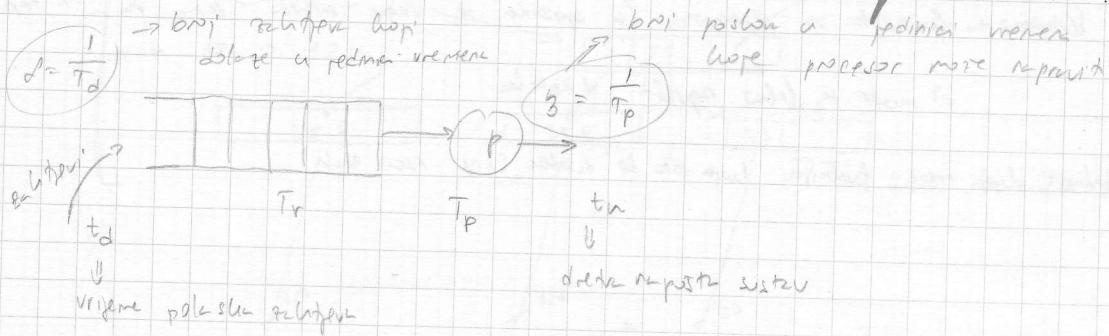
C-LOOK

$$POMAN: (98-25)+21$$

- C-SCAN

$$POMAN: (95-25)+21$$

7. Analiza vremenih putova i računarske



$$T = t_d + T_p + T_r \Rightarrow \text{vrijeme radnog vremena u sustavu}$$

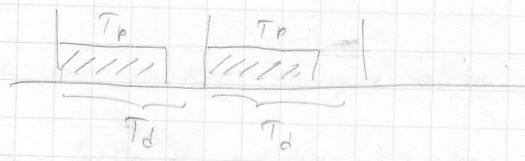
$$T = T_p + T_r$$

$T_p \Rightarrow$ vrijeme procediranja

$T_r \Rightarrow$ vrijeme očekivanja u redu

- deterministički sistemi \Rightarrow predviđeni scenarij \Rightarrow sve se odvija u određenim vremenima
- nedeterministički sistemi \Rightarrow npr. nekonvergentne aplikacije

Deterministički sistemi



\rightarrow reči T_d dodata novi zadaci, zatim u procesor

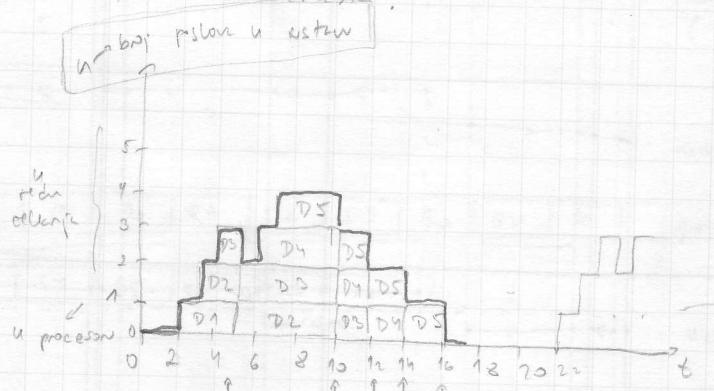
$$\beta = \frac{T_p}{T_d} \Rightarrow$$
 izkoristivost procesora

$$\beta = \frac{\beta}{\beta} \leq 1$$

Primer A.1.

Pripremati se da se izvrši 5 poslova kopija u njeni sledećim periodima, s $t=20$ jedinicama vremena.

Posao	t_d	T_p
D1	2	3
D2	3	5
D3	4	2
D4	6	2
D5	8	2



$$n = \frac{T_1 + T_2 + T_3 + T_4 + T_5}{20} = \frac{20}{20} \bar{T} = 1 \cdot \bar{T}$$

broj poslova u jedinicu vremena

$$\bar{n} = 1 \cdot \bar{T}$$

Littlesov pravilo

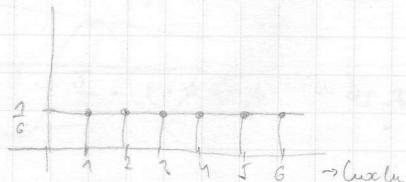
vrši operaciju

Neketerministični sistem

Vrijednost događaja

- Števina vrijedna : - diskretna (brojevi: 1, 2, 3, 4, 5, 6) \Rightarrow broj zbiljnih
 - kontinuirana (temperatura) \Rightarrow neizjač obnade

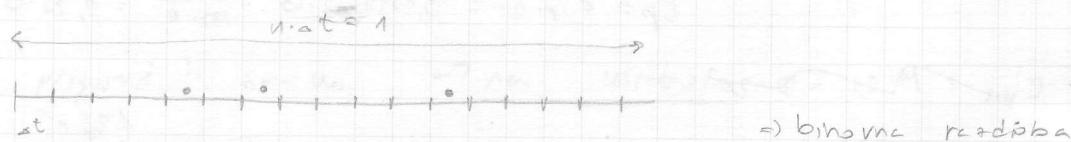
Raspodjelja vjerjetnosti:



$$\int = 1 \quad (\text{priroda može biti i euklipsna vjerjetnost})$$

- $\left[\begin{array}{l} \text{ne učinku} \Rightarrow \text{poissonova raspodjela} \\ \text{ne učinku} \Rightarrow \text{eksponentijalna raspodjela} \end{array} \right]$

Poissonova raspodjela



$$p+q=1 \rightarrow \text{du je date}$$

vrijednost da su date u nekom st

$$b(u, n, p) = \binom{n}{u} \cdot p^u \cdot q^{n-u}$$

- ① Kolika je vjerjetnost da se od 10 bacajuca kocka dobije 3 rezke?

$$\binom{10}{3} \cdot \left(\frac{1}{6}\right)^3 \cdot \left(\frac{5}{6}\right)^7$$

Vjerjetnost da nema rezke: $b(0, n, p) = q^n = (1-p)^n$

Vjerjetnost da u sustavima vidi od 3 događaja: $b(3, n, p) = 1 - (1-p)^n$

Poissonova raspodjela $\Rightarrow n \rightarrow \infty$ (st $\rightarrow 0$)

$$\boxed{p \cdot n = \lambda} \Rightarrow \text{konstanta} \quad (p \rightarrow 0)$$

$$b(0, n, p) = (1 - \frac{\lambda}{n})^n \cdot \left(\left[1 + \frac{1}{-\frac{\lambda}{n}} \right]^{-\frac{n}{\lambda}} \right)^{-\lambda} = e^{-\lambda} \quad | = p(0, \lambda)$$

$n \rightarrow \infty, \lambda \rightarrow 0, p \rightarrow 0$

Nastavak ... OS \rightarrow Nedeterministički sistem

$$b(0, n, p) = e^{-\lambda} = p(0, \lambda)$$

$$p(u, \lambda) = ?$$

$E \rightarrow$ očekivanje (prosječna vrijednost)

$$\begin{aligned} p(u, \lambda) &= \frac{(u)^r}{u!} r^u \lambda^{u-r} \\ p(u-1, \lambda) &= \frac{(u-1)^r}{(u-1)!} r^{u-1} \lambda^{u-1-r} \end{aligned}$$

$$E = E(u) = \sum_{u=0}^{\infty} u \cdot p(u, \lambda) = \lambda$$

$$\Rightarrow \lambda = \lambda$$

$$p(u, \lambda) = \frac{\lambda^u}{u!} e^{-\lambda}$$

O u jednoj minuti pojavljuje se 120 objekata na istoj lokaciji po vremenskoj

distribuciji sljedećoj sekundi pojavljuje se 200 objekata?

$$u = 2 \quad \lambda = \frac{120}{60} = \frac{12}{6} = 2$$

$$p(u, \lambda) = \frac{2^2}{2!} \cdot e^{-2}$$

interval vremena (broj vremenskih jedinica)

$t = \text{trajat}$

$$p(u|t, \lambda) = \frac{(\lambda t)^u}{u!} e^{-\lambda t}$$

$$p(0, \lambda) = e^{-\lambda t}$$

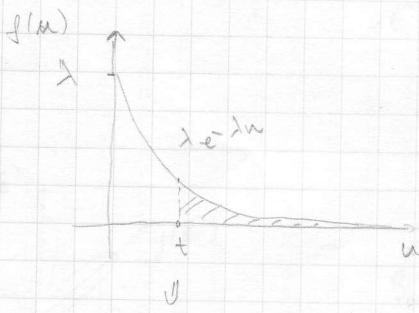
= vjerojatnost da se ne dogodi ni jedan
objekt u svaku u vremenu t

$$p(120, \lambda) = 1 - e^{-\lambda t}$$

\checkmark vjerojatnost da će biti 1 objekat objekt

EXPONENCIJALNA SADRŽABA

- nešto druga gledana je na ulazu (golub ne može preći)



$$\int_t^{\infty} \lambda e^{-\lambda u} du = e^{-\lambda t}$$

\Rightarrow vjerojatnost da će ne dogodi dogoditi u t

TRAJANJE ANEŠE DVA OBRAZLA POSLUJU U SUSTAVU

ako je $T_d > t \Rightarrow$ nije se desilo obrazlo u t

ako je $T_d < t \Rightarrow$ može se desiti da je obrazlo u t

$$E(t) = \int_0^{\infty} u f(u) du \sim \frac{1}{\lambda} = \frac{1}{L} \Rightarrow \text{verjene predi da dolazi}$$

	Poisson	Exponential
Vrijednost da nema nijednog događaja u t	e^{-t}	e^{-t}
Vrijednost da je događaj u t	$1 - e^{-t}$	$1 - e^{-\lambda t}$
prosječan broj dolazaka	$\lambda = t$	$\frac{1}{\lambda} = \frac{1}{t} = T_d$

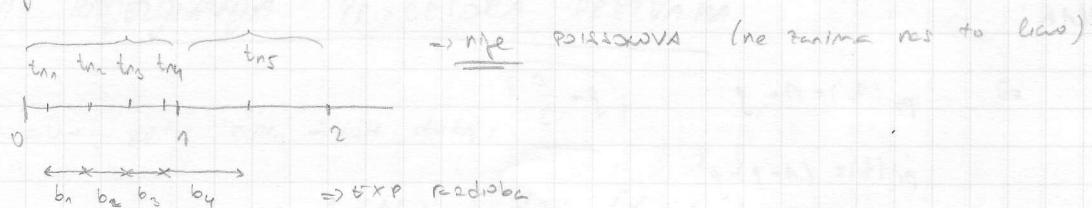
Akutne infekcije i Biliomni rizikom dolazak i sljepenječkom

Rizikom obrađe

Modeliranje dolazaka



Modeliranje odlažaka

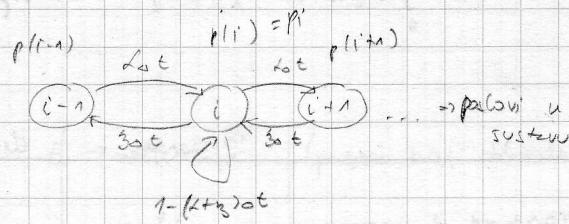
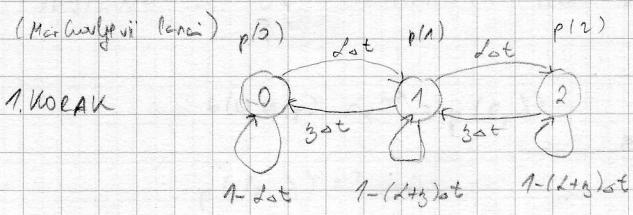


$$p(U(\lambda t) = 0, L) = e^{-\lambda t} = 1 - \lambda t + \frac{(\lambda t)^2}{2!} \approx 1 - \lambda t \Rightarrow \text{ok je ne dovoljno njen}$$

$$p(U(\lambda t) > 0, L) = \lambda t \Rightarrow \text{bar 1 je dovoljno}$$

⇒ analogno: $1 - \lambda t$, λt

Wolken je weroerdost de n systeem me i poslave? $(p(i)) = ?$



✓
Weroerdost poslave je jedny stany u druge

2. KORAK

$$p_i(t+\Delta t) = [1 - (\lambda + \mu)]p_i(t) + \lambda \cdot p_{i-1}(t) + \mu \cdot p_{i+1}(t)$$

$$p_0(t+\Delta t) = [1 - \lambda \cdot t] \cdot p_0(t) + \mu \cdot p_1(t)$$

3. KORAK

$$\frac{dp_i(t)}{dt} = \Rightarrow \text{stationarni stanje} \quad (\text{Weroerdost je konstantne})$$

$$\frac{p_i(t+\Delta t) - p_i(t)}{\Delta t} = \left| \Delta t \rightarrow 0 \right| = \frac{dp_i(t)}{dt} = 0 = -(\lambda + \mu) \cdot p_i(t) + \lambda \cdot p_{i-1}(t) + \mu \cdot p_{i+1}(t)$$

$$\frac{p_0(t+\Delta t) - p_0(t)}{\Delta t} = -\lambda \cdot p_0(t) + \mu \cdot p_1(t) = 0$$

$$(\lambda + \mu) \cdot p_0(t) = \lambda \cdot p_{i-1}(t) + \mu \cdot p_{i+1}(t)$$

$$p_0(t) = \frac{\mu}{\lambda + \mu} p_1(t)$$

$$p_i(t) = \frac{\lambda}{\lambda + \mu} p_{i-1}(t) + \frac{\mu}{\lambda + \mu} p_{i+1}(t)$$

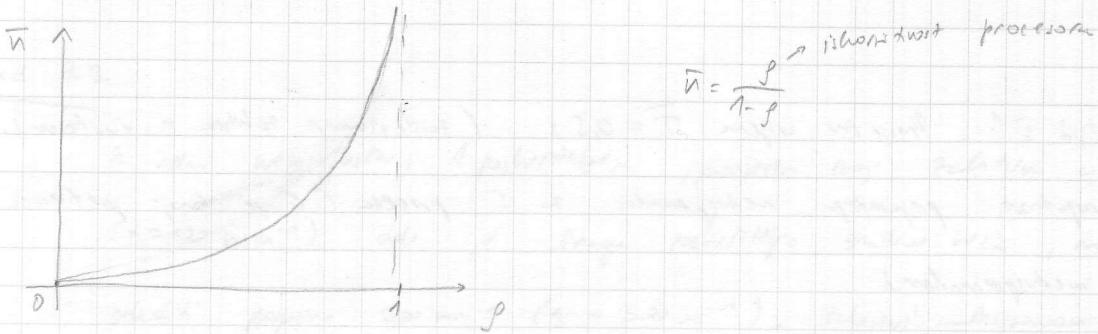
4. KORAK

$$\Rightarrow p_0(t) = 1 - \rho \quad \boxed{\rho = \frac{\lambda}{\lambda + \mu}}$$

$$p_i(t) = (1 - \rho) \cdot \rho^i$$

$$\boxed{\bar{n} = \sum_{i=0}^{\infty} i \cdot p_i(t) = \frac{\rho}{1-\rho} = \frac{\lambda}{\lambda - \mu}}$$

$$\boxed{\bar{T} = \frac{\bar{n}}{\lambda} = \frac{1}{\lambda - \mu}}$$



- u determinističkom sustavu: $p \in [0, 100\%]$

- u redeterminističkom sustavu: $p \rightarrow 1 \quad n \rightarrow \infty \Rightarrow$ sustav je raspodijeljen

* auto pre svih biti na blizu 100%

Primer 7.2.

Priprezatimo se za potrebe i potražio konvenciju prema ograničenju spremnika
kada je ustop od N-potencije. U sustavu se tako može reći:

Naprve M-a u N-i. Koliko rezervi sustava moraju imati RAM-a, da bi
se osigurala iskoristnost procesora? Koliko je vjerojatnost prepunjenja spremnika?

$$p((i > N)) = \sum_{i=N+1}^{\infty} p(i)$$

$$p(i > N) = 1 - p(i \leq N) = \sum_{i=0}^{N} p(i) = (1+p) \sum_{i=0}^{N} p^i$$

zbog po N, i u ne M
u $p(i \leq N)$

$$= (1+p) \cdot p^N \sum_{i=0}^{\infty} p^i = (1+p) \cdot p^N \cdot \frac{1}{1-p} = p^{N+1}$$

vjerojatnost prepunjenja
spremnika

NACINI PODJEGIVANJA PROCESORA DREVAMA

→ čime postige npr. 2 vrste drevi:

- crni poslovni (L_u, S_u)

- dugi poslovni (L_d, S_d)

Koji su parametri sustava $(L, S, p) = ?$

$$\begin{aligned} L &= L_u + L_d \\ \frac{1}{S} &= \frac{1}{L_u} + \frac{1}{L_d} = \frac{S_u + S_d}{L_u + L_d} = \frac{\frac{L_u}{S_u} + \frac{L_d}{S_d}}{L_u + L_d} = \frac{\frac{L_u}{S_u} \cdot \frac{1}{1} + \frac{L_d}{S_d} \cdot \frac{1}{1}}{L_u + L_d} \end{aligned}$$

$\boxed{p = p_u + p_d}$

2nd 7.1.

$\lambda = 2 \text{ s}^{-1}$. Projekcija vrijeme $\bar{T} = 0,5 \text{ s}$ (zadnjim razdoblju u sustav).

Uspoređujemo period početne redispersione sa 5 periodom ($5 \text{ je broj petina u redispersioni}$)

$$p(i>N) = p^{N+1}$$

$$p = \frac{\lambda}{\bar{T}}$$

$$\bar{T} = \frac{1}{\lambda - \lambda}$$

$$\lambda - \lambda = \frac{1}{T} \Rightarrow \lambda = T + \frac{1}{T}$$

$$p = \frac{\lambda}{\lambda + \frac{1}{T}} = \frac{2}{2+2} = \frac{1}{2}$$

$$p = \left(\frac{1}{2}\right)^{5+1} = 2^{-6}$$

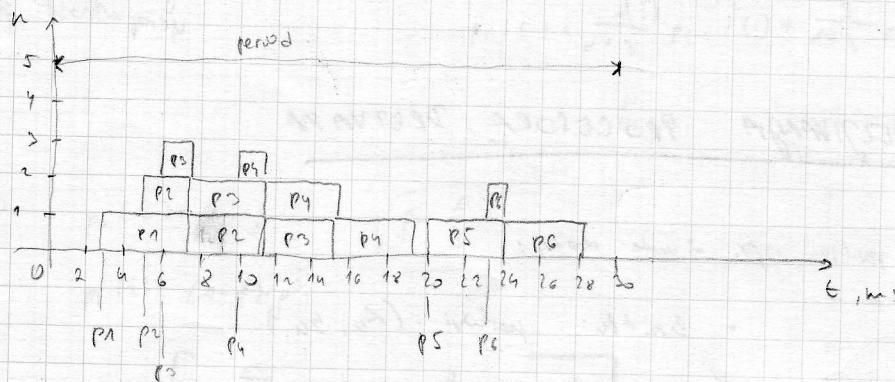
2nd 7.2.

U nekom determinističkom sustavu postoji 6 periodički mijenjajućih

vrijednosti i to P_1 u 3 ms, P_2 u 5 ms, P_3 u 6 ms,

P_4 u 10 ms, P_5 u 20 ms, P_6 u 23 ms. Svi postovi traju isto

$\Rightarrow P_i = 6 \text{ ms}$. Projekcija redisponirajućih postava u sustav (\bar{T}) i projekcija broja
postava u sustav (\bar{n}).



$$\bar{T} = \frac{4+6+10+9+10+20+23}{6} = \frac{82}{6} = \frac{32}{3} = 6,162 \text{ ms}$$

$$\bar{n} = \lambda \cdot \bar{T} \quad \lambda = \frac{6}{30} = \frac{1}{5}$$

$$\bar{n} = \frac{1}{5} \cdot \frac{32}{3} = \frac{32}{15}$$

$$\bar{n} = 2,13$$

#	t_p	T_p	t_n	T
P_1	3	4	7	4
P_2	5	4	11	6
P_3	6	4	15	9
P_4	10	4	15	9
P_5	20	4	24	4
P_6	23	4	28	5

Zad 2.B.

Ježeli wab system s 1 połwintelem, prędkość ruchu zwiększa się o minutę je 100.

($\Delta = 100 \text{ m} \cdot \text{s}^{-1}$) dalej, iż średnia połwintelna średnia wega, mówiąc odrędu prędkości $320 \text{ m} \cdot \text{s}^{-1}$. Zapiszmy - Równanu, obracać binarne.

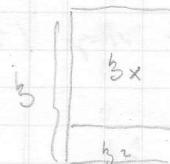
Wolność ruchu najpierw połwintelu połwinteliszu времена mówiące rezerwem:

jeżeli czasu, a da wykonać, daje się odczepić prędkościowe wiele godzin

$$\Delta t = \frac{120}{60} \text{ s}^{-1} = \frac{2}{3} \text{ s}^{-1} \quad t = \frac{320}{60} = 5 \text{ s}^{-1}$$

$$\bar{T}_2 = 2,3 \quad \bar{T} = \frac{1}{\frac{1}{3}-\frac{1}{5}} = \frac{1}{\frac{2}{15}} = \frac{15}{2} = 7,5 \text{ s},$$

$$\frac{\bar{T}_2}{\bar{T}} = \frac{2}{7,5} = \frac{20}{75} = 0,67 \text{ puta}$$



$b \rightarrow$ połwinteliszu wiele = 5 połon u sekundzi

$$\bar{T}_2 = \frac{1}{b_2 - \Delta} \Rightarrow \frac{1}{\bar{T}_2} = b_2 - \Delta \Rightarrow b_2 = \Delta + \frac{1}{\bar{T}_2} = \frac{5}{2} \text{ s}^{-1} + \frac{1}{2}$$

$$\frac{b_2}{b_1} = 1 - \frac{b_2}{b_1} = 1 - \frac{\frac{13}{2}}{5} = \frac{17}{30} = 56,67\% \quad b_2 = \frac{5}{2} + \frac{1}{2} = \frac{12}{6} = 2,167 \text{ s}^{-1}$$

Zad 2.4.

Ježeli wab system s 1 połwintelem prędkość ruchu zwiększa się o sekundę

je 100 ($\Delta = 100 \text{ s}^{-1}$). Szybkość zwiększa się (21, 22, 23). Ježeli

obracać prędkościowe 5 ms, 22 8 ms, 23 10 ms. Wobólto po

połoteli zwiększać 21 30%, 22 40%, a 23 30%, odręduje pojęcie

$$\Delta = 100 \text{ s}^{-1}$$

$$\begin{array}{ll} 21 & T_{p1}=5 \text{ ms} = \frac{1}{31} & 30\% \\ 22 & T_{p2}=8 \text{ ms} = \frac{1}{32} & 40\% \\ 23 & T_{p3}=10 \text{ ms} = \frac{1}{33} & 30\% \end{array}$$

wielkość czasu, tj. prędkościowe wiele zwiększenia położenia w systemie, tzn. prędkość dalej u systemu malejąca i od 10 zwiększa.

$$\bar{T} = \frac{1}{\frac{1}{3}-\frac{1}{33}} = \frac{1}{\frac{2}{32}} = \frac{32}{2} = 16 \text{ s}$$

$$d_1 = \frac{1}{T_{p1}} = \frac{1}{5} \text{ s}^{-1} \quad d_1 = 30 \text{ s}^{-1}$$

$$d_2 = \frac{1}{8} \text{ s}^{-1} \quad d_2 = 40 \text{ s}^{-1}$$

$$d_3 = \frac{1}{10} \text{ s}^{-1} \quad d_3 = 30 \text{ s}^{-1}$$

$$\bar{T} = \frac{1}{\frac{1}{3}-\frac{1}{33}} = \frac{1}{\frac{2}{32}} = \frac{32}{2} = 16 \text{ s}$$

$$\frac{1}{\bar{T}} = \frac{d_1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3} = \frac{1}{31} + \frac{1}{32} + \frac{1}{33}$$

$$\bar{T} = \frac{32}{\frac{1}{31} + \frac{1}{32} + \frac{1}{33}} = 33,4 \text{ ms}$$

$$\frac{1}{\bar{T}} = \frac{3}{10} + \frac{4}{8} + \frac{3}{10} = \frac{3}{2} = 1,5$$

$$\frac{1}{\bar{T}} = \frac{3}{2} + \frac{16}{32} + 3 = \frac{72}{32} = 2,25 \text{ s}^{-1} \Rightarrow \bar{T} = \frac{32}{2,25} = 14,3 \text{ ms}^{-1}$$

$$n > 10$$

$$p(i>10) = p^{10+1} - s^{10} = \left(\frac{2}{3}\right)^{11} = \left(\frac{100}{129,83}\right)^{11} = \underline{5,6\%}$$

2d 2.5.

Under polvirketiden er der nemlig et andet redt forvirketid.

Vedvirkningen på de 3 polvirketider redt i forløbene nedenfor oplyses her.

$$P_1: \alpha_1 = 20 \text{ min}^{-1} = \frac{7}{6} \text{ s}^{-1}, f_1 = 20\% \text{ (process effektivitet)}$$

$$P_2: \alpha_2 = 200 \text{ min}^{-1} = \frac{20}{6} \text{ s}^{-1}, f_2 = 200\%$$

$$P_3: \alpha_3 = 150 \text{ min}^{-1} = \frac{15}{6} \text{ s}^{-1}, f_3 = 10\%$$

P₃ på processrækkens nedenst. 50% mindre end P₁, 100% mindre end P₂.

Frægmættet produktivitetsværdi (F) udgøres af de tre virkninger parallelt

na P₃.

$$\beta_1 = \frac{\alpha}{f} = \frac{7}{6 \cdot 0,2} = \frac{12,2}{12} = \frac{35}{6} \text{ s}^{-1} \quad d = d_1 + d_2 + d_3 = 420 \text{ min}^{-1}$$

$$\beta_2 = \frac{20 \cdot 12}{6 \cdot 2} = \frac{240}{12} = \frac{120}{6} \quad d = \frac{420}{6} = 7 \text{ s}^{-1}$$

$$\beta_{31} = 1,5 \cdot \beta_1 =$$

$$\beta_3 = \frac{15 \cdot 10}{6 \cdot 1} = \frac{150}{6} = \frac{25}{3} \quad \beta_{32} = 2 \cdot \beta_2$$

$$f_3' = f_3 + f_{32} + f_{31} = f_3 + \frac{f_2}{2} + \frac{f_1}{1,5} = 0,1 + 0,15 + 0,2 \cdot \frac{2}{3} = \frac{1}{6} + \frac{4}{15} = \underline{38,3\%}$$

$$F = \frac{1}{3-d}$$

$$\beta = \frac{1}{f_3'} = \frac{7}{0,383} = \underline{18,272}$$

$$\overline{T} = \underline{88,7 \text{ ms}}$$

Ponavljanje za 21

① U postavom stanju korišće u strukturi + linijski fiksni mehanizam.

U neliči program raspoređuju se očekve u buferu i pohranjivo

stranice 5, 3, 2, 1, bitovi pristupa (access) 0, 1, 0, 1, kroz koja

pločaju se u 9. okviru. Neka su ovaj redoslijed u spomeniku: 5, 2, 3, 8.

(program im 10 str)

str	A
0	5 0
1	3 1
2	2 0
3	1 1

a) tablica pristupa

str	okvir	bit pristupa
1	3	0 0
2	2	1
3	1	1
4		0
5	0	0 1
6		0
7		0
8	3	0 1
9	0	0 0
10		0

5 1	5 1
3 1	3 1
2 0	2 1
1 0	1 0

5 1	5 0
3 1	3 0
2 1	2 0
1 0	8 1

→ Budući da str. veruju RAM-u orden se

ustrojila po mreži (često je A bit mijenjan)

②

NTFS - 10 MB deštive velike podjeljene po na stranice veličine 4 KB.

Prvi 4 MB pomoći od blok LCN = 34562, sljedeći 2 MB

pomoći od LCN = 12345, te zadnje 4 MB od LCN = 20000.

Naprednji tablica pristupa,

VON	LCN	# (broj stranica)
0	34562	1024
1024	12345	512
1536	20000	1024

$$\therefore 4 \text{ MB} = 4 \cdot 1024 \text{ KB} = 1024$$

(3)

Sinhronizaci: dretre wojt ture wod (H₂O).

Monitor M

mt br [2];

#define U 0

ted upfetz ulc[2];

#define H 1

(slipage;

void* atom(X)

// X i H li O

{

water_clock(&M);

br(x2++);

while (br[M] < 2 || br[O] < 1) \Rightarrow n realizate fuziune (ne de acord se pișă)

cond-wait (ulcs[X], &M);

br_atom++;

also if (br_atom < 3)

{

cond-wait (slipage, &M);

y more {

slipage - molitate;

cond-broadcast (slipage);

}

br_atom--; // Să 3 dătre slipage sau'n 1

also je (br_atom == 0)

{

br[H] = br[U] = 0;

cond_signal (ulcs[H]); $\times 2$

cond_signal (ulcs[O]);

water_clock (&M);

y

(4)

U există 3 dătre și molitate 1 și 3 wojt atunci zedni' luc

ține și dătre realize u redni' BSEM(1); Poate urjednost OSEM(1) $\neq 1$.

Aceea se u tot creatur păzire posturi-betem(1) neajuns să re

șe sprijini' ne zednu, și u redni' prioritati'. (3 și majorat)

dodata (1) {

dodata (1) {

celyj-BSEM(1);
celyj-OSEM(2);

ispis(i,"K.O.");

postavi-BSEM(1);

also je (i ne parna)

postavi-OSEM(2);

ispis(i,"K.K.O.");

a

b

c

y

j

R&D	BSEM	R&D OSEM	PREPRAVNE	ISPIS	OSEM.V	BSEM.V
3 2 1	-	-	3a	3 K.O.	0	0
2 1	-	-	3b 2a		0	0
1	-	-	3c 2a	3. K.K.O.	1	0
3 1	-	-	2a	2 K.O.	0	0
1	-	-	3c 2b		0	0
1	3	3	2c	2 K.K.O.	0	0
2 1	3	-	-		0	0

→ potpis: Zeclo

→ mv <old><new> → move dat (nazv rename)
↓
move bit i dat file i direktorij

mv dmes.txt prav.txt → move dat file

→ rm <file> → remove (kazne se brise)

- BASH ciklusi:

- tab → dopuni imena fajlova rezultata prema dat file, tab dušljivo nudi sve dat s ovim istim nazivom
- history → ispis redosleda komandi

PREDVODIČ

\n → prvi buffer razmaka pred u printf-a
- gcc kompilator

gcc <ime>
↳ napisati a.out dat file → izvršni dat file

• /a.out → pokretanje (potrebno je napisati ophvat path)
✓ simbolni imenici
trenutna direktorija

-gcc priv.c -o priv (priv će bit .out)

- debug : GDB

PREDVODIČ program: CTRL+C (^C)

ZAVRŠAVANJE program: CTRL+Z

fg → program se može pozvati zavrsavanju program
↳ foreground

AKTIVNI procesi: PS → print status

KILL <PID> → ubije aktuelni proces
-9

- u pico - u see first file deletions \Rightarrow copy automatically

\rightarrow past \rightarrow doesn't like .mif2

\Rightarrow make "program"

SCP \rightarrow secure copy

WINSCP \Rightarrow presentation of filetree in windows in Linux + обратно

Logout

- ipcs.

häufige Fehler:

- line.C file me.cpp \Rightarrow C++ program

- bus error segmentation fault; core dumped \Rightarrow pointer

char * \Rightarrow nix define address is 4

<int *? = <char *? \Rightarrow error trying to point to address ac 4

- socket

`Sleep(12);`

`x=10`

`while (x) x=sleep(x);`

`sigset (SIGNAL, prejedna_routine);`
funkcija

- ine funkcije \Rightarrow adresa \Rightarrow prve intruzije funkcij

3. less

3.c

čvorac \rightarrow potnik (min 3 & max broj čvorova)

$M \geq K$ učvora

- skoristi pomočna vrste 2 s, a temeljna vrsta 1s
- dolaze skozno nov. & rješ. D oblik
- čvorac ne potekne D oblik



Vježba 2A MI

1 MI. 2013. (B)

(1) bee shop a prioritní příkaz

$p_1 \text{ a } 0; 2 \text{ ms}$

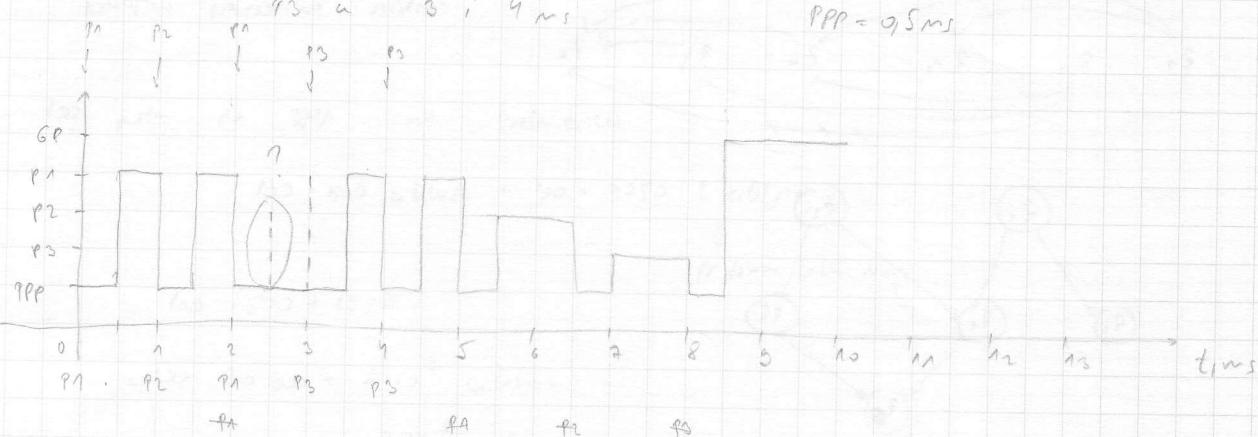
$p_1 > p_2 > p_3$ (prioritet)

$p_2 \text{ a } 1 \text{ ms}$

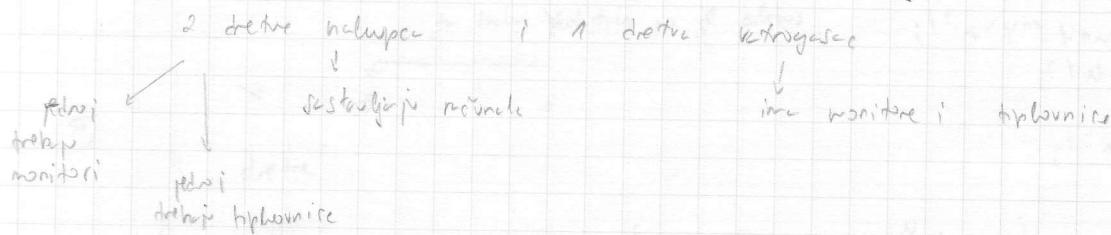
$p_1 = 1 \text{ ms}$

$p_3 \text{ a } 3; 4 \text{ ms}$

$PPP = 9,5 \text{ ms}$



(5) (nejtěžší funkce)



Halváče 1 ū

dru (1) ū

reží - BSEM (monitor);

uzaví - monitor;

postavi - BSEM (PEZAN - STOL);

suství - rečnice;

Halváče 2 ū

dru (1) ū

reží - BSEM (TYPLOVNICA);

uzaví typlovnici;

postavi - BSEM (PEZAN / TYPLOVNICA);

suství - rečnice;

y

Výkrojář ū

dru (1) ū

reží - BSEM (PEZAN - STOL);

postavi - BSEM (PEZAN - STOL);

BSEM (PEZAN - STOL); $v = 1$;

postavi - BSEM (TYPLOVNICA);

inov postavi - BSEM (MONITOR);

y

BSEM (PEZAN - STOL); $v = 1$;

BSEM (MONITOR); $v = 0$;

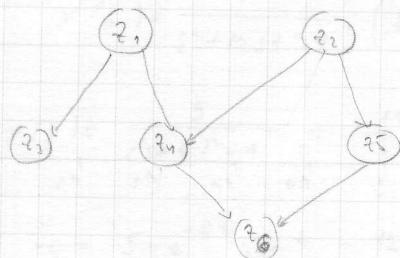
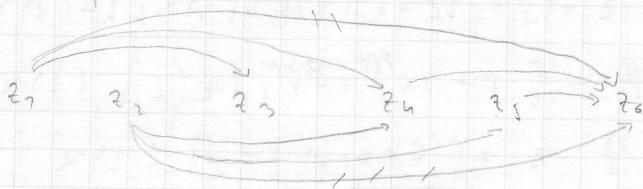
BSEM (TYPLOVNICA); $v = 0$;

1. Ml. 2012.

⑤

$$z_1 \rightarrow z_2 \rightarrow z_3 \rightarrow z_4 \rightarrow z_5 \rightarrow z_6$$

	z_1	z_2	z_3	z_4	z_5	z_6
M_1	D	D		K		K
M_2	K		K			
M_3		K		D	D	K
M_4				K	D	



⑥

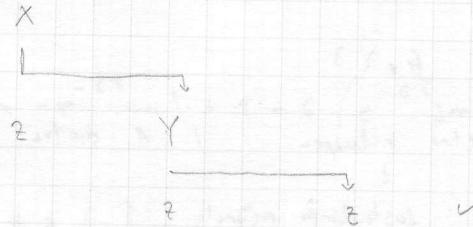
```
printf("X\n");
```

```
if (fch() == 0) {
```

```
    printf("Y\n");
```

```
    fch();
```

```
y  
printf("Z\n");
```



⑦

$$z_1 = z_2 = 0$$

$$\text{PRW} = 1$$

Wch

$$j = 1 + (2 - 1) = 2$$

$$\text{BL} = ?$$

$$j = 1 + (2 - 1) = 2$$

$$z_1 = 1;$$

$$z_1 = 0;$$

```
du( z2 = 1 & & PRW = j );
```

$$\text{PRW} = j;$$

y

y

- obj my ist vorneis b7 u k.o.

(14)

$$f = 10 \text{ MHz} = 10^7 \text{ operacija u sekundi}$$

975 000 rivoči

1 "programirano" rivoč je prekidač max 2750 rivoča

DMA - dobiti sučin 2. sabirnički ciklus

j-fra sprednji UI → 0 ciklusa

j-fra Prelid-UI → 200 sabirničkih ciklusa

stvarno vrijeme =?

postrojba procesora vrijeme?

 $\rightarrow 100 \text{ puta de DMA između Prelid-UI}$

$$100 \cdot 500 \text{ ciklusa} + 100 \cdot 9750 \cdot 2 \text{ ciklusa}$$

pridobivati sučini drugi

$$= 100 (500 + 19500)$$

$$\approx 100 \cdot 20000 = 2 \cdot 10^6 \text{ ciklusa}$$

$$t = \frac{2 \cdot 10^6}{10^7} = \frac{2}{10} = 0,2 \text{ s} \quad \checkmark$$

$$\rightarrow \text{procesorsko vrijeme} \Rightarrow 975 000 + 100 \cdot 500 = 1 025 000 \text{ ciklusa} = 0,1025 \text{ s}$$

pridobivati sučini po 1 ciklusu

(15)

4. drezne

ulazna

obraditev

obraditev-UV

spremi

relaj-BSEM(NMV);

dohvat slube;

postavljanje slube;

postavljanje-BSEM(N);

postavljanje-BSEM(UV);

relaj-BSEM(N);

dohvat;

postavljanje-BSEM(N-GOTOV);

relaj-BSEM(UV);

obraditi;

postavljanje-BSEM(UV-GOTOV);

relaj-BSEM(NMV);

relaj-BSEM(UV-GOTOV);

slupi reži;

postavljanje-BSEM(NMV);

obraditi;

nahraniti;

zavrsiti radni akutacija alarm;

BSEM(N), $\varphi = 0^\circ$ BSEM(UV), $\varphi = 0^\circ$ BSEM(NMV), $\varphi = 90^\circ$ BSEM(N+GOTOV), $\varphi = 0^\circ$ BSEM(NUV-GOTOV), $\varphi = 90^\circ$ \rightarrow treba bolje

(18)

Student {

```

    ūčenje_BSEM(ko);
    ūčenje(5 minuti);
    ko = 1;
}

```

if rad i zadache;

```
postavi_BSEM(ko);
```

```

for (i=0; i<n; i++) {
    postavi_OSEM(pravila);
}

```

```

for (i=0; i<n; i++) {
    ūčenje_OSEM(ko);
}

```

Oponi ispite i objavi rezultate;

```

for (i=0; i<n; i++) {
    postavi_OSEM(resultati);
}

```

grupe 6.

(18)

Student {

```

    postavi_OSEM(pravila);
    ūčenje_OSEM(pravila);
}

```

rešenji ispit;

```
    ūčenje_BSEM(ko);
```

else (kaošno == NE) {

```

        ūčenje_BSEM(ko);
        n++; } predvi_ispit();
        postavi_BSEM(ko);
    }
}

```

predvi rezultate;

```
    Inace postavi_BSEM(ko);
```

}

OSEM(pravila) > 0;

BSEM(ko).v = 1;

OSEM(pravila).v = 2;

OSEM(resultati).v = 0;

Student {

```
    ūčenje_BSEM(ko);
```

else (kaošno == NE) {

n++

```
    postavi_BSEM(ko);
```

đelaj_OSEM(pravila);

rešenja();

```
    postavi_OSEM(kaošno);
```

```
    ūčenje_OSEM(kaošno);
```

}

Inace postavi_BSEM(ko);

}

Kaošno je global varijable

kaošno = 0;

(?)

→ ne koristi se
BSEM sustav

d neki sustav

zapis studente

koji rešuju o redni

BSEM

Student {

```
for (i=0; i<n; i++)
```

ūčenje_OSEM(pravila);

predi zadatake;

```
for (i=0; i<n; i++)
```

postavi_OSEM(pravila);

đelaj_60_min;

đelaj_BSEM(ko);

kaosno = DA;

```
    postavi_BSEM(ko);
```

Oponi ispite i objavi rezultate;

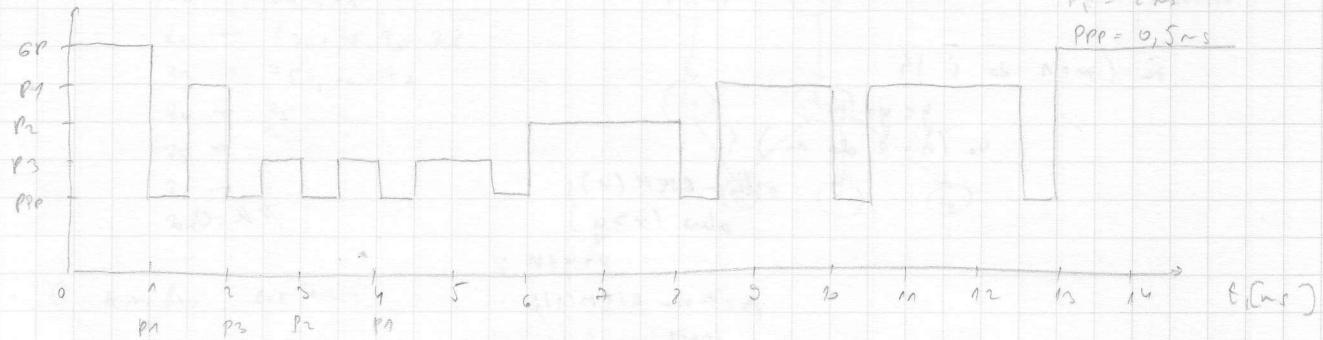
```
for (i=0; i<n; i++)
```

postavi_OSEM(resultati);

}

kaosno = NE

⑥



$$P_3 > P_2 > P_1$$

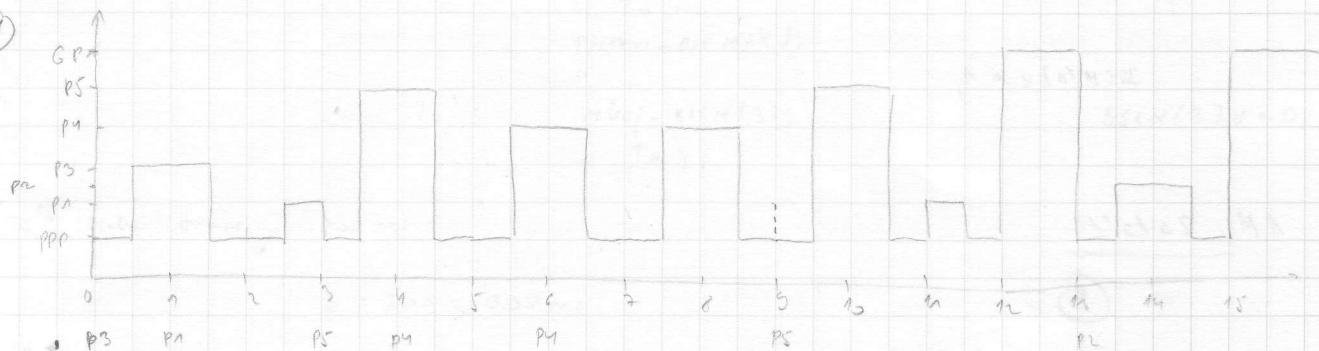
$$P_1 = 2 \text{ ns}$$

$$\text{PPP} = 0, 5 \text{ ns}$$

\Rightarrow P1 durchsetze in höheren PP

1. All 2007102

①



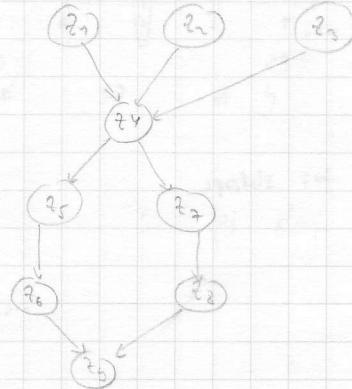
$$PS >$$

$$P_1 = 1 \text{ ns}$$

$$\text{PPP} = 0,5 \text{ ns} \quad \text{sc: 1000 ns}$$

②

- $z_1 \rightarrow z_4, z_6, z_8$
- $z_2 \rightarrow z_4, z_6$
- $z_3 \rightarrow z_4, z_8$
- $z_4 \rightarrow z_5, z_7, z_8, z_6$
- $z_5 \rightarrow z_6, z_8$
- $z_6 \rightarrow z_7$
- $z_7 \rightarrow z_8$
- $z_8 \rightarrow z_9$
- z_9



⑤

do (1) {

$a = 1;$
SWAP (a, z);

do ($a == 1$) SWAP (a, z);

k.O.

$z = 0;$

N.W.O.;

a	1
1	1

1

z	1
1	1

1

0	1
1	1

$\rightarrow a$ ist lokale Variable

6

Dreton (i) 5

\hat{z}_m ($m=1$ do i)

$$y = y + m$$

$$\forall n \ (n = 1 \text{ do } m) \ \{$$

ókai - BSEM (1);

also ($x > y$)

$$x = y + n ;$$

posterior - sternum;

. 188

$$y = y - n;$$

$\text{BETM}(n), v = 1,$

1. MI: 2012/11

6

6

P1 >

$$|P_i| = \alpha \omega$$

bz shape

$$|PPP| = 0,5m$$

$$t = 6 \text{ ms}$$

$$T_P = 2$$

$$k_2[1] = 0$$

$$k_-\bar{z}[z] = 0$$

$$k_2 [3] = 1$$

$\text{won}[1] = \text{win}$

$$W_N(\gamma) = \{s, \cancel{\gamma}(s)\}$$

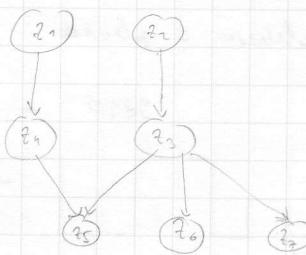
$\text{kon}[\beta] = \text{ny definer}$

(?)

immiss

(2)

$z_1 \rightarrow z_4, z_5$
 $z_2 \rightarrow z_3, z_6, z_7$
 $z_3 \rightarrow z_5, z_6, z_7$
 $z_4 \rightarrow z_5$
 $z_5 \rightarrow$
 $z_6 \rightarrow$
 $z_7 \rightarrow$



• G dešavljanje BSEM

T_1, T_4, T_5

$T_1' : T_1;$

postavi - BSEM(1);

BSEM(1).v = 0;

$T_4' : \text{deluj - BSEM}(1);$

$T_4;$

postavi - BSEM(4);

BSEM(4).v = 0;

$T_7' : \text{deluj - BSEM}(3);$

$T_7;$

BSEM(3).v = 0;

c) svaki trapez 200 ms

$$3 \cdot 200 = 600 \text{ ms}$$

(3)

jež s
deluj - BSEM(1);
ispis(s);
s++;
postavi - BSEM(1);

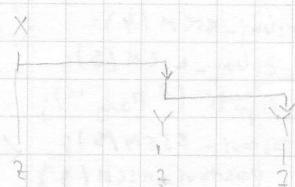
korenjava s
deluj - BSEM(1);
deluj - BSEM(1);
ispis(s);
s++;
postavi - BSEM(1);
postavi - BSEM(2);

na s
deluj - BSEM(2);
deluj - BSEM(2);
ispis(s);
s++;

BSEM(1).v = 0;
BSEM(2).v = 0;

na poj. dej

(4)



{1, 2, 3}

(14) $f = 10 \text{ MHz} = 10^9$ ciklusa u sekundi.

985 000 ciklusi

$\rightarrow 100 \text{ puta DMS redi na 1 sekundu}$

$\rightarrow \text{DNA dobija svaki d. ciklus u sekundi}$

Peklud-UI = 500 ciklusi

priklovnjene = $100 \times 500 + 985 000 \times 2 = 50 000 + 1970 000 = 2020 000$ ciklusi

\downarrow
100 puta
preko

svaki drugi ciklus

$$t = \frac{2020000}{10^2} = 0,2020 \text{ s}$$

procesorska vrijeme = $500 \times 100 + 985 000 = 50 000 + 985 000 = 1035 000$ ciklusi

$$t_p = 0,1035 \text{ s}$$

(15)

dok(1) {

dohvati instrukciju; $\langle PC++ \rangle$;
dohoditi i dohvati operande;
izvrši instrukciju;

ako je postavka prekida {

onemogući prekidanje;
prijesti u fuzivni niz reda;
stavi PC i SR u staj;
u PC stavi adres obrade prekida;

}

(16)

. Ping Ping Pong Pong

Ping }

dok(1) {

$B=6;$ } celijski_BSEM(2);
 } celijski_BSEM(1);
 } ispis("Ping");
 $A=A+2;$ } postavi_BSEM(1);
 } postavi_BSEM(A);

}

Pong }

dok(1) {

celijski_BSEM(4); $\langle A=0 \rangle$;
 } celijski_BSEM(3);
 } ispis("Pong");
 } postavi_BSEM(3); $\langle B=B-2 \rangle$; (4, 2)
 } postavi_BSEM(B);

}

BSEM(n).v=1;

BSEM(2).v=1;

BSEM(4).v=1;

BSEM(3).v=0;

31

