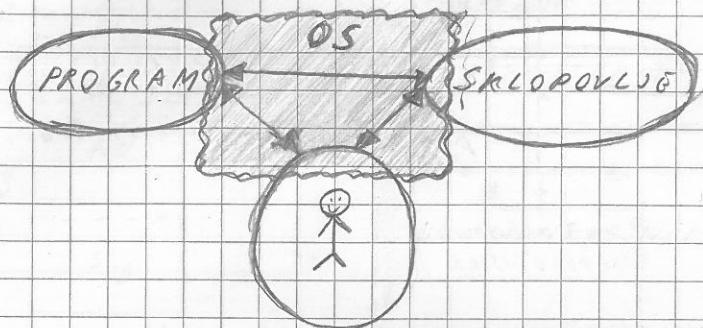


! OPERACIJSKI SUSTAV JE SKUP PROGRAMA KOJI NAM
OBLAKŠAVAJU RAD NA RAČUNALU



! ZADACI OPERACIJSKOG SUSTAVA

1. OBLAKŠATI RAD NA RAČUNALU
→ TO RADI TAKO DA STVARA
SVA DO DEBE

2. DJELOVANJE ISKORISTAVANJE
SVIH DJELOVA RAČUNALO
→ OV OS SVI ŽIVI VJEĆI U RIM

3. VISJE PROGRAMSKI RAD
→ VISJE PROGRAMA ODJEDNO
→ DOS NIJE JAKOV

{#include <stdio.h>

```
int main (void)
{
    printf ("Hello. world!")
    return 0;
}
```

! NAPOMENA: KODOSTOJE "n"
NA KRAJU

ZAŠTO?

"n" PREDNI MEDIUSPAĆNIK

! OSNOVNI POJMOWI:

→ PROGRAM JE SASTOJINI NIZ INSTRUKCIJA

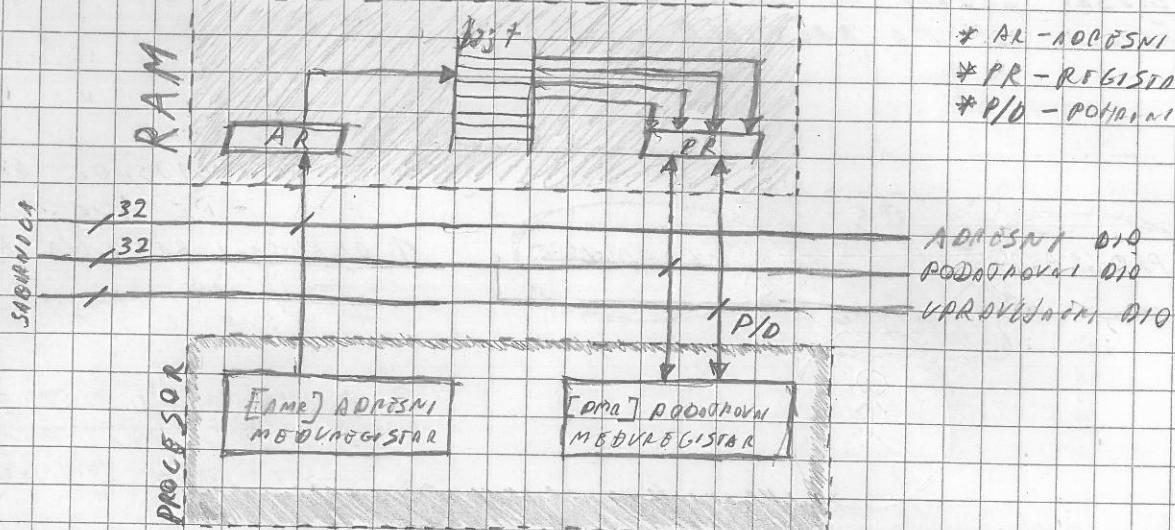
→ PROCES JE SVO OVO ŠTO JE POTREBNO
DA BI SE PROGRAM MOGAO IZVODITI
Proces je skup računalnih resursa
koji omogućuju izvođenje programa

• SVAKI PROCES SE SASTOJI OD
BOČOM JEĐOVI DEFTVO (THREAQ)

→ DRŽIVA (THREAQ) JE NIZ INSTRUKCIJA
U IZVODENJU

→ KONTROLI DRŽVO JE SASTOJINA SVIH
RJEŠIĆARA

MODEL JEFTONOSTAVNOG RAČUNALA



* AR - ADRESNI REGISTAR
* PR - REGISTAR SADRŽAJA
* P/SW - POZICIJSKI REGISTAR

ADRESNI DIO
PODATAKOVNI DIO
UPRAVILJACI DIO

! PROCESOR JE DEFINIRAN S:
 1. SKUPOM REGISTARA
 2. INSTRUKCIJSKIM SKUPOM

- ! OSNOVNI REGISTRI:
1. ADRESNI MEĐUREGISTAR
 2. PODATKOVNI MEĐUREGISTAR
 3. PROGRAMSKO BROJILO (PC)
 4. REGISTAR STANJA (ZASTAVICE)
 5. REGISTAR KONTROLNE STOLICE
 6. INSTRUKCIJSKI REGISTOR
 7. SKUP REGISTARA OPTO NOMJENE

ponavljanje

dohvat instrukciju na koju pokazuje PC;
sjekodizaj instrukciju;

PC++;

operande dovedi na ALU;

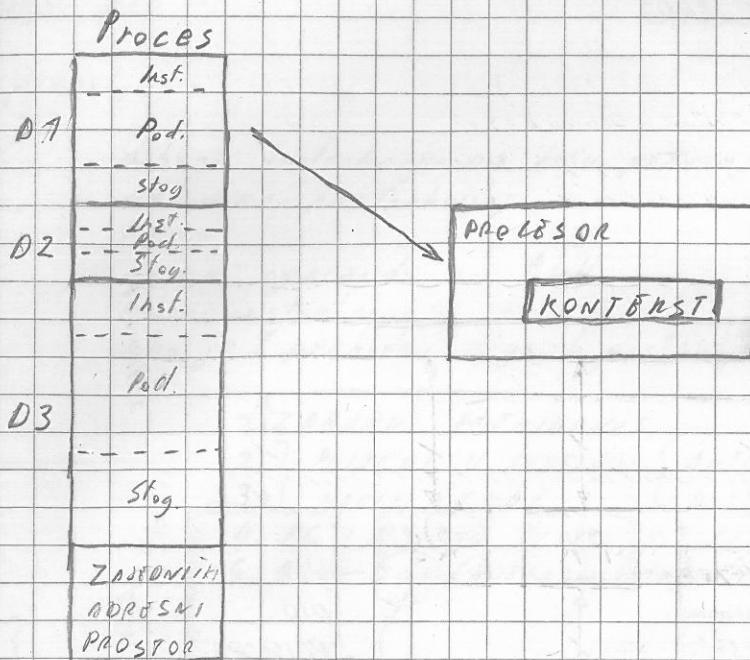
izvrši instrukciju;

} sve dok IMA STRUJO;

|| sljedeća instrukcija

- SKUPovi INSTRUKCija:
- 1. za premještanje podataka
- 2. za obavljanje operacija
- 3. programski skripti ili grananja
- 4. posebna upravljanja pješčovnja

► PRIMJERI INSTRUKCija U UDŽBOVIM IRU



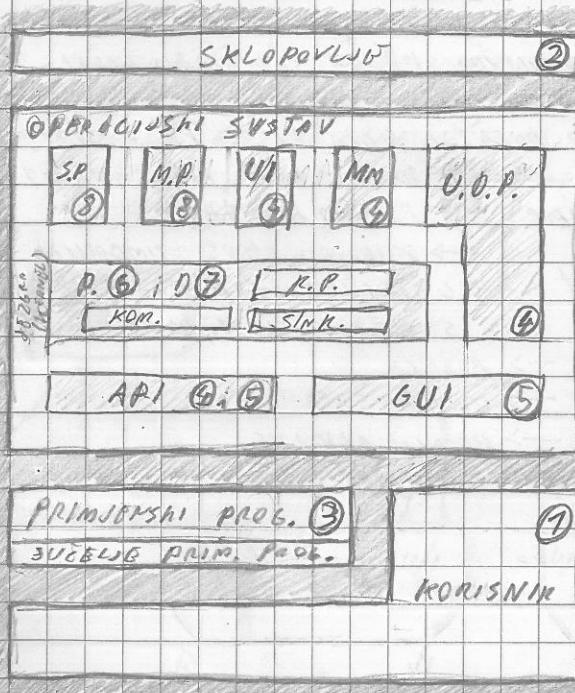
* KONFILKST UZIMA SVĘ
POZISIJE, ALEI POSEBNO
JAO FILIP:

- REGISTAR STAVUR (REGISTR)
- PC (ZADANIE VMAIS.)

! ADRESNI PROSTOR PROCESA
PODIJELJEN JE NA DVE TROJICE
POZISIJE I ZASLONNIKI
ADRESNI PROSTOR

==> DREVNI PODPROSTOR

- INSTRUKCIJE
- PODACI
- STOG



U/I - UPRAVljavanje

MM - MEMORY MANAGEMENT

U.O.P. - UPRAVLJANJE DATOĆÖNIM
PODSISTEMOM

API - APPLICATION PROGRAMMING
INTERFACE

P. - PROCESI

D. - DRŽAV

GUI - GRAPHICAL USER INTERFACE

KOM. - KOMUNIKACIJA

SINR. - SINKRONIZACIJA

R.P. - RASPREDJIVAJE POSLOVA (BALANSIRANJE)

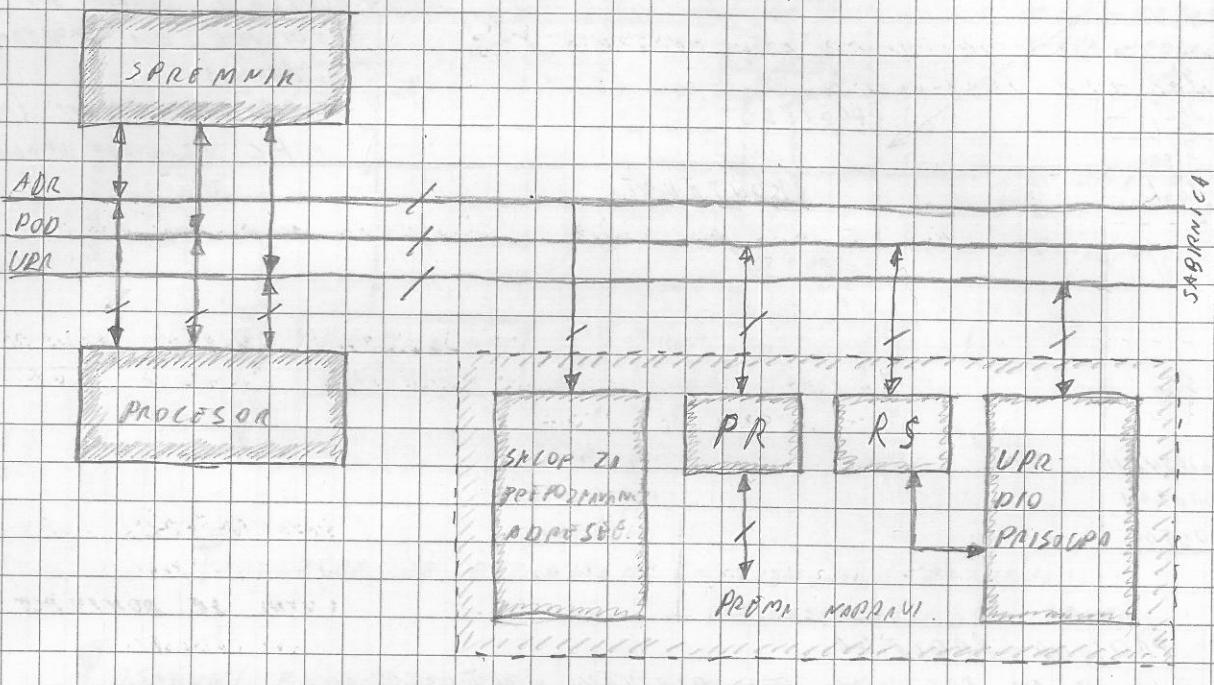
M.P. - MOĆNI PODSISTEM

S.P. - SIGURNOSNI PODSISTEM

DJELOVI OPRECOJSKOG SUSTAVA

* JOŽEGMA (KOMP), OSNOVNO R.P. SE BRINE NAKOJI SMO DAVOVI
* VIŠEDRŽAVNI POSTOJIĆOMO MIJENJANJUMLAZNIĆOM

OBAVLJANJE U/I OPERACIJA, PRENIJEM RAD



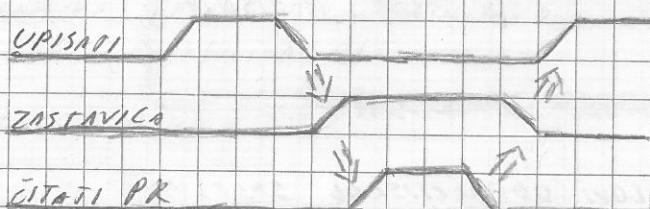
čitanje RS;
dok je (zastavica == 0); // RADNO ČEKANJE!
čitanje PR;

- ! INSTRUKCIJSKI SKUP:
- LDR R0, [R1] → načini R0 sa saopšnjom košt
je na adresi R1
 - STR R0, [R1] → prenosi sadržaj s R0 u SPREMINIK
na adresu na koju počinje R1
 - ADR R1, simboličko ime → PSOU00 NAOZ030
→ 0010000 10000 simboličko imo
 - CMP R0, #vrijednost → USP0001 5000000 + R0
S NOVOM VRIJEDNOSTI
 - BEQ simboličko ime → BRANCH
→ UVJETNI SROK
 - BAL simboličko ime → BREMI ALVOS
→ BESLUZOMI SROK

ADR R1, RS
ADR R2, PR

PETLJA: LDR R0, [R1]
CMP R0, #0
BEQ PETLJA

LDR R0, [R2]



* U ovom načinu rada procesor 99,9% vremena čeka, a 0,1% radi

(*) PROTOKOL DVOJICNOG RVNOVANJA - SYNCHRONIZACIJA U/I MNOZIVE S PROCESOROM
(čime su UPISATI i ZASTAVICA)

PREKIDNI RAD

ponarkejaj

{ dekvati instrukciju na koju pokazuje PC;
dekvadiraj instrukciju;

PC++;

dovedi operande na ALU

izvrši instrukciju;

ako je (proshodni signal postavljen i dozvoljeno je prekidanje)

{ ZABRANI PREKIDANJE

2. PRIZEDI U SUSTAVSKI { AKTIVIRAJ SUSTAVSKI ADRESNI PROSOD

3. NAOJN RODA { AKTIVIRAJ SUSTAVSKU KAZALJNU STOGA

4. PC I RS NA SUSTAVSKI STOG

5. $PC \leftarrow 3dt$ (POTPROGRAMA)

} }
→ OVO SE NALazi U NENOM
OD SAKRIVENIH REGISTARA

* PRAVA NAPRODA POTPROGRAMA
JE SPOMONUE PORIFERIJSKI
(OVAJ JE PONIJETO
SVI REGISTRI OSIM PC I RS)

! VAZITI SE U PREDHODNU DACTU

1. AUTOMATSKI PROSESOR POCASI U NAOJN
RODA PREDVINKO DODOVE

2.3. AKTIVIRAJ SE ODGOVARAJUĆI (KONTROLNI
ILI SUSTAVSKI) REGISTRI KOZALJNI
STOGA, SPOMNIČKI PROSTORI

4. DOZVOLI PREKIDANJE APO SU PREDV
BIU DOZVOLJENI U PREDVINKO DACTU

5. OBNOVI SODRŽAJ PREGRAMNEGA BREVLA (PC)

* ROD PREKIDA ZAVISI

1. OBNAVI NODENAST

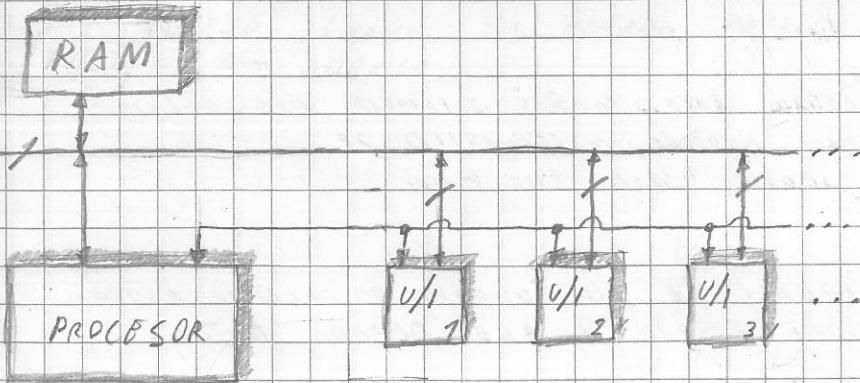
2. OBNOVI RS

3. OBNOVI PC

* U VROM NOJINGA PADA

98% VOJNOVA SE
KODISNO ISKOZILOM

62% 000 NO „PRIMNI POSLOV“



* PREKIDNA VELJAČA

- JEDNA ZADÖMISLJENA JEKE

=> SAKRJENI ILL

* VREDITI SE U PREDHODNU DACTU * VREDITI SE U PREDHODNI NAOJN RODA

VREDITI RS,
OMI DOSEN
VREDITI PC

VREDITI SE U KONTROLNIH NAIJIN NODP
(SLEDEĆEM POKUŠAJI DA SE SAKRJUJU
KONTROLNIYI PROTOKOL PROSODA I PREDVINKA SODE)

PREKIDNA RUTINA - PROGRAMSKO RJEŠENJE

- STAVKA UPUTA PODATAKA: T_P (TEKUĆI POKRIVAC)
(OPERACIJSKOG SUSTAVA) KON[n] (POJAS KONTROLNIH ZASTAVICA)
K_Z[i] (KONTROLNO ZASTAVICO)
L> BILO JEDNO JE LI SE DOGODIO POKRIVAC
I JEL GA ODGOVOR

V KAO ZA PREKIDNU RUTINU

spremiti kontekst na sustavski stog; // osim PC, SR
ispitnim lancem utvrditi vratak prekidač, tj. index pristupa I;
// moguće provjeravaće ispravnost prekidač
oznaka ičekanja $K_Z[I] = 1$;
ponistiti zastavicu u registru stanja pristupa I;
dok je ((postoji $K_Z[J] \neq 0$) && ($J > T_P$))

{
odabri se vrijeli J ; // max oznaka ičekanja $\neq 0$
 $K_Z[J] = 0$;

pohraniti kontekst sa sustavskog stoga i T_P u $KON[J]$;
 $T_P = J$;

omogućiti prekidanje;

obraditi prekidač J ;

zadržati prekidanje;

sa $KON[J]$ obnovi kontekst na sustavski stog i T_P ;

} obnoviti kontekst sa sustavskog stoga; // osim PC i SR

omogućiti prekidanje;

vratiti se u način rada prekinute akcije i obnoviti PC;

/* PREKIDNA RUTINA

* prije izvođenja:

** onemogućiti prekidanje

** PC stavi na sustavski stog

*/

* Ako stavljaju se integer umjesto bit, može zapamtit uši pokrida

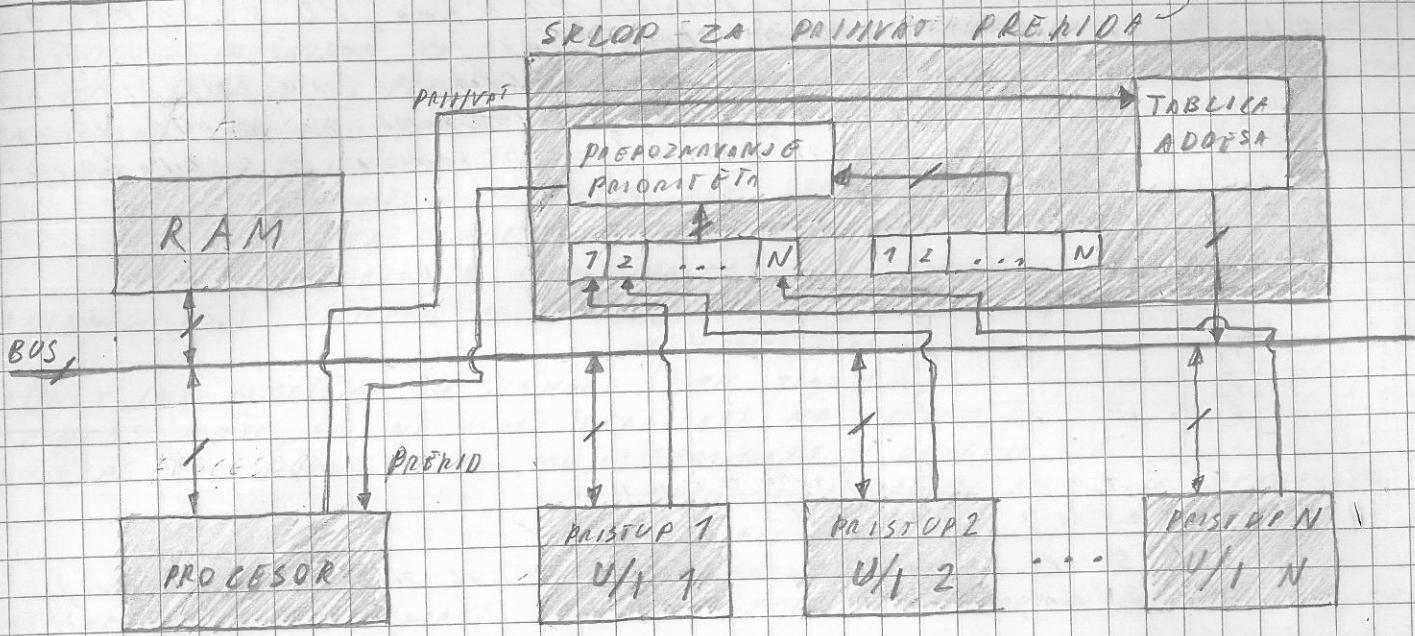
=> NO, TO SE U PRIMPI NEMOGUĆI, NEISPONOVIVO JE

=> PAMOI SE SAMO JEDAN, OSTALO TAKO ŠAVIĆI

! U PROGRAMSKOM RJEŠENJU PREKIDNOG PODSUSTVA PREGIDNA RUTINA
"POZIVA SE" NAO SE DOGODI POKRIVAC I KAO OBRAĐA NOVOG
PREGIDAČA ZAVRŠI

SKLOPOVSKA POTPORA PREKIDONU

50 STRANCI 10
DIO ADRESA



U SUSTAVU SRLOPA ZA PRIMAVAT PREKIDA KULOMSKI POSLOVI SO OBRAVLJAJU SVAKI PRVI RAD OBRAVA NERAD PREKIDA ZAVRSI I ONDA NODA SJ DOGODOK PREKIDA I VISE FAZING CO TBKUIEG PREKIDA.

zko je (prihvata signal postavljen i omogvito je prehoduje)

1. zatrži prekidanje
2. zhtiviraj sustavski spremnički prostor
3. zhtiviraj sustavsku kredaljku stoga
4. počekaj CIJELI KONDEKT uz sistemski stog
5. POSTAVIO SIGNAL 'PRIMAVAT'
6. SA SABIRNIĆE PROVODEM SADRŽAJ I 12 NRŠBA 00000101 ADRESU PUNKCIJU ZA OBAVLJU PREKIDA
7. PC ← ADRESA

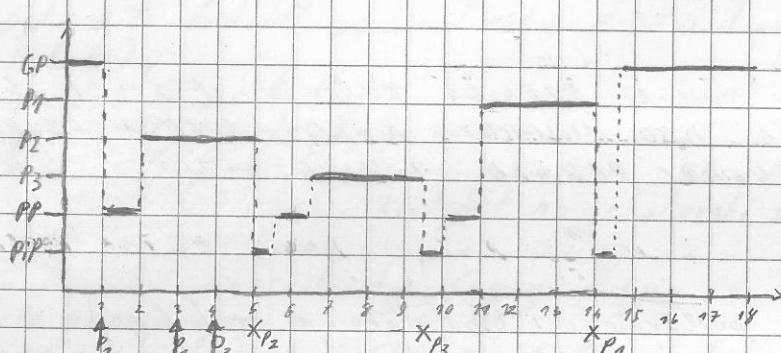
}

ZAD. 3.1. U NEKOM SUSTAVU JAVLJAJU SE PREDICI P₁ U 3. ms, P₂ U 7. ms, P₃ U 10. ms PRIORITET PREDICI 00000001 JE BROJEN (P₃ IME NARVOVI PRIORITET). OBROD PREDICA TANJ 3 ms ($\Delta P = 3$ ms). GRANIČNI PREKIDAČI DUGOTRUDNOST PROCESORA U GLAVNOM PROGRAMU (GP) PREDVODIĆU ZA OBROD PREDICA (P_i) I PRED PREDVODIĆU ZA PRIHVAT PREDICA (PP) I PREDVODI 12 PREKIDA (PIP) I TO.

d) U IDEALNOM SLUČAJU

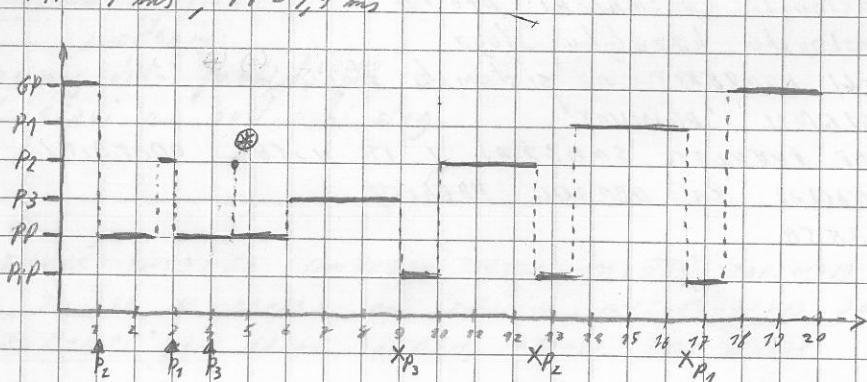


b) BEZ SKLOPA ZA PRIHVAT PREDICA (OBROD UZ ZABOONJENO PREDVODIĆE)



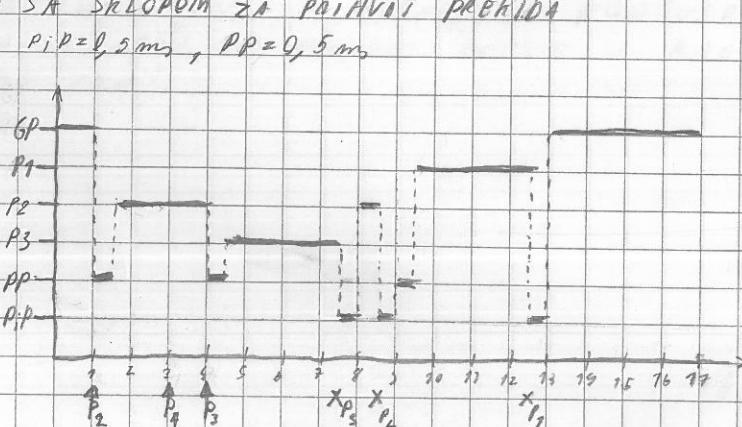
* ovo je u smislu
sustavu
=> nejednici alarmi

c) BEZ SKLOPA, ALI S PROGRESONIM PREDVODIĆEM



⊕ PRIMIO PREDVODIĆE NIJE
BIO DOZVOLJEN, A
KAO SMO GA POZVOLILI
SHVAPILI SMO DA SI
DEŠIO JOS JEDAN PREDICID

d) SA SKLOPOM ZA PRIHVAT PREKIDA



SIGNALI

- ANALOGIJA PREDSTAVLJENA NA ROZINI PROCESORA (SIGNALI SU DIO OS-a)
- U OBOJU SLOVIMA, I PROKLA I SIGNAL SLUŽI ZA OBRAĐUJU IZNIMNIH SITUACIJA
- NE SIGNALI PREDSTAVLJU PROCESI
- SIGNAL ŠTO JE: 1. JEZBRA OS-a
2. PROCES MOŽE POSLATI SOM SEBI SIGNAL (NPR. SIGALARM)
3. NEKI DRUGI POLOŽI SJEDJE SIGNAL
4. KORISNIK (NPR. CTRL+C)
- SIGNALI KOJI UKIŠTAJU PROCESE - SIGKILL (9)
- SIGTERM (15)
- UNIXIMA POISČENO 30 SIGNALA (POPIŠE signal.h)
- PROCES MOŽE, ALI NE MOŽE REAGIRATI NA SIGNAL
- PROCES KHO DOBIDE SIGNALNE MOŽDJE: 1. PRIMJEROTO I OBRAĐOVA GA
2. PAZHIVAT I OBRAĐA DOPOLNIT PUNAKLJOM
3. ZADRŽI SIGNAL
4. IGNORIRAJ GA
- SIGNAL NE SAZRŽI NIŠAVKE DODATNE INFORMACIJE (BODI SIGNALA JE INFORMACIJA)
- OS PAMTI SMOJE JEDAN SIGNAL
- PUNAKLJE MOŽE POAISTIĆI: int sighold(int sig);
int sigrelse(int sig);
- NOVI SIGNAL SE MOŽE PREMVAJATI SMOJE VREĆICE SA ISOLIRANOM OBRAĐUJU
• OS PAMTI SMOJE PO 7 SIGNALA ZA SVAKI MOJES

```
int t;  
sleep(100);  
t=700;  
while(t) t=sleep(t);
```

#OVO ČE SIGNALA SPAVAT 700ms
PAZI SA SLEDEĆIM

* KAO SB PROGRAM IZVODI POSLEDNJE

⇒ PROCESIMA SVOJ PID

(TO MOŽE APROVIZIRATI PROGO PS -a)

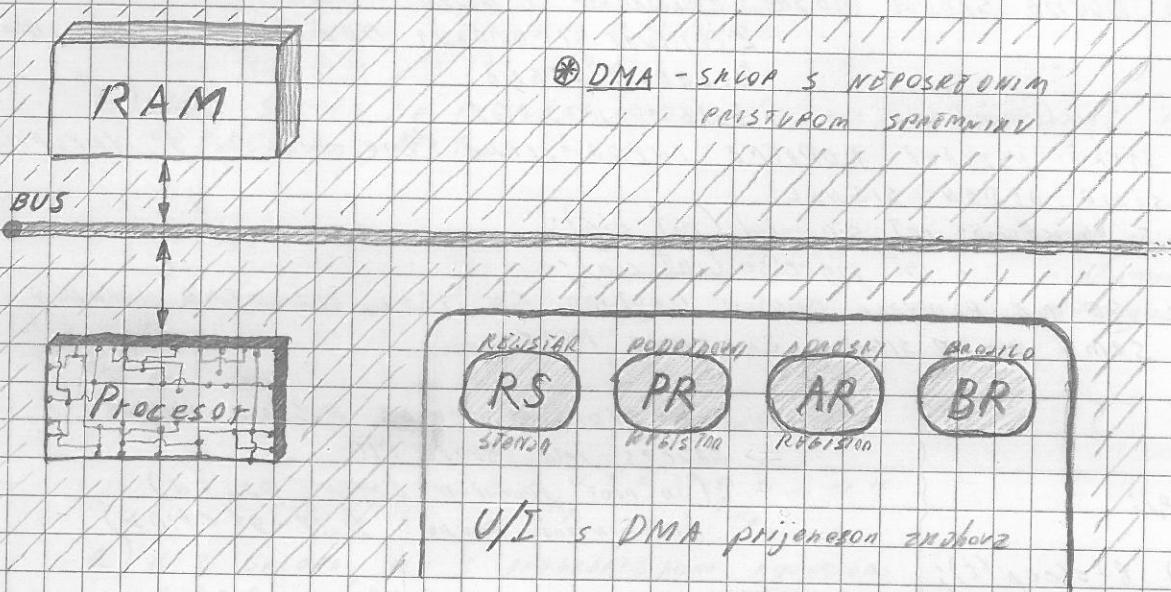
(TEOMINIJAN POGAO: kill -9 <PID>)

PREKIDI GENERIRANI UNUTAR PROCESORA

• PROCESOR ŠALJE PREKIDE:

- [] 1. ZBOG NEVRENE MERA GROSJE / NPA. DODJELJUJE S NOVOIM /
- [] 2. ADRESIRANJE NEPOSEDOVOG MEMORIJELOVOG LOKACIJE / BUS ERRORE /
- [] 3. DEKODIRANJE NEPOSTUPNOG IZINSTAVLJAJUĆEG KODA ili
 - [] • POSTOJI INSTRUKCIJA ZA PROGRAMSKO IZAZIVANJE PREKIDA
 - DA BI SE POZVOLIO JEZGRINE PONAVLJAJE

DMA, APDENOS BLOKOVA ZMENJENA



dok je (BR) \neq 0

zatvori sabirnicu;
čekaj na dodjelu sabirnice
postavi adr dio sabirnice na AR;
prenosi znak iz ihi u PR;

AR++;

BR--;

}

postavi signal PREKID

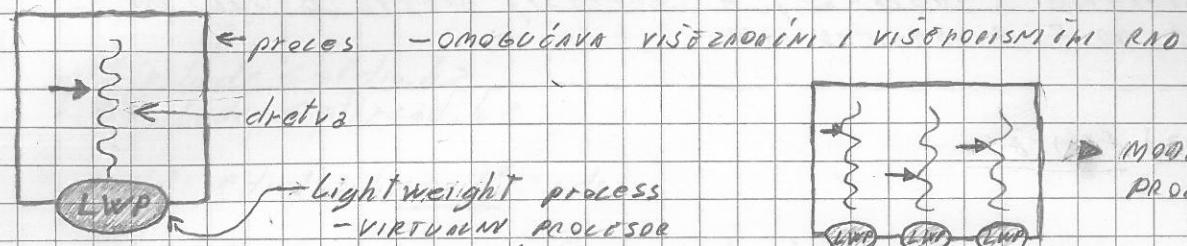
→ PROCESOR JE 99,99% VREMENI
KORISNO AKTIVAN

→ PRIMJER: TVAOI DISK
(NAPREDOVANJE VREDNINA)

④ SLIČNO DMA POSTOJI DODJELJIVANJE SABIRNIČA I ZA VISE PROCESORA
- PROCESOR MOŽE I NE MORI TAKOVI SAGURNJUĆI

MEDUSOBNO ISKLJUČIVANJE U VIŠEOPRETVENIM SUSTAVIMA

• PROCES



VIŠEZADACI RAD

ubaza dretva	A	B	C	D	...
radna dretva	A	B	C	...	
izlazna dretva	-	-	A	B	...

• POKRETANJE VIŠE PROCESA

- PROCES ROĐITELJSKI => PROCES DIONIĆE
- DIONIĆE JE KOPija ROĐITELJSKE
- PODACI NISU ZAŠTOMIĆI
- DVA PROCESA PO DEPOZITU NOMENOVI
NISTI ZAŠTOMIĆI
- # DRETVOIMA U ISTOM PROCESU
JE SVE ZAŠTOMIĆI
- PROCES ROĐITELJSKI MORA ZAVRŠITI
NARAVNO ŠTO ZAVRŠI PROCES DIONIĆE
- PROCES BJEZ ROĐITELJA: SIROĆ
↳ OPERACIJSKI SUSTAV SE
BLINIĆE ZA NUGA
- PROCES ZOMBIE
↳ PROCESA MEMA VIŠE, ALI JE
PREKONTRALNO ZAPISAN U
OPERACIJSKOM SUSTAVU

int fork (void); // stvara novi proces

!-> -1 => greška

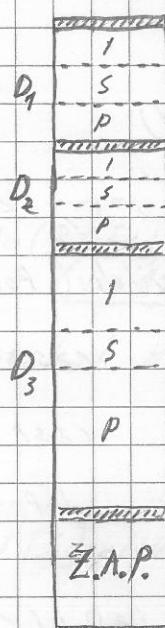
!-> 0 => proces dioniče

!-> PILO održi je => procesu rođitelju

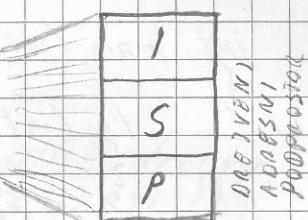
int pthread_create (kazaljka, kazaljka, kazaljka na dretvu, kazaljka na argumente);
NULL

* kazaljka - void ponovljivo

Proces



Dretva



DEJSTVENI
ADRESNI
PODPROSTOR

• PROCES SE STVORA

U POČETKU S JEDNOM
OPITVOM => GLAVNOM DRETVOM

thread_create - RAZLIČICE SO NA SVIM UNIX-u
pthread_create - STANDARDIZOVANI POSIX NAčIN STVARANJA DRETVA
create_thread - WINDOWS

↳ RAZLIČICA NA FUNKCIJO (IMO PUNKCIJI)
= RAZLIČICA NA PAKV INSTRUKCIJU

ZAD. NAPISATI PROGRAM KOJI OSTVAREO VISOZADOĆNI RAD UZ
POMOĆ PROCESA /OPSTVI/. I JESEN I DRUGI PROGRAM
STVORJU 3 INSTANCI, A PROCESI /OPSTVI/ BEEVOG OO
I DO 5 I ISPISUJU IH.

2) PROCESI

```
#include <stdio.h>

void proces(int i)
{
    for (int j=0; j<=5; j++)
    {
        printf("Proces: %d, proces %d \n", i, j);
        sleep(1);
    }
}

int main(void)
{
    for (int i=0; i<3; i++)
    {
        switch (fork())
        {
            case -1: printf("Greška! \n");
                        exit(0);
            case 0: proces(i);
                        exit(0);
            default: break;
        }
    }
    for (int i=0; i<3; i++)
        work(NULL);
    return 0;
}
```

* PROCESI SU KOPIRAJE ČITAVOG OVOG KODA

* DVEGA DOBJAVI JEH I PONAVLJAVI NA NOJI DIO KODA JE PREDLOZIO STAO
* exit() JE NUŽEN DA DVEGA NE PONAVI PO KADU I PREDIJO NOVI PROSES

b) DRETVA

```
#include <stdio.h>
#include <pthread.h>

void *dretva (void *rdr)
{
    for (int i=1; i<=5; i++)
    {
        printf("Dretva: %d, probaz %d\n", *(int *)rdr, i);
        sleep(1);
    }
}

int main(void)
{
    int i, BR[3];
    pthread_t t[3];
    for (i=0; i<3; i++)
    {
        BR[i]=i;
        if (pthread_create(&t[i], NULL, dretva,
                          (void *) &BR[i])) {
            printf("Greska!\n");
            exit(0);
        }
    }
    for (i=0; i<3; i++)
        pthread_join(t[i], NULL);
    return 0;
}
```

- MORA BITI VOJNO KAZALJKA

* MORA BITI BR[], A NE i, JER BI IMAO SVO PROVOD
POKAZIVAC NA ISTI BROJ (čest bi bio broj koji je os viđen
kao nazivni parametri provoda)

* DRETVA JE KOPija koda u void *dretva

! KOD GLAVNE DRETVE ZAVRŠI, ZAVRŠI I PROCES

=> DA NEĆE pthread_join sve provode biti u redovu poze

* PROVOD OBILO SVO UNUTAR ZADATKOVNIH PROSTORA

=> ZATO PROVOD MOGU PRISTUPITI int BR iz glavne dretve

! S KOD MEMORIJISKIM LOKACIJAMA IL KOSTIMA PROVODA ITAK NEMOĆE SE
DOMONDA, A S KOD NEMOĆNIH LOKACIJAMA U PAJE DATUM
POJEDINOG NEMOĆA SE KODOMA.

! ZADATKI: svaki proces ima glavnu boštu už svoj prostor

UVJETI NEZAVISNOSTI ZADATAKA (ORETVI)

$$K_i = f_i(D_i)$$

$$K_j = f_j(D_j)$$

$$\left/ \left(D_i \cap K_j \right) \cup \left(D_j \cap K_i \right) \cup \left(K_i \cap K_j \right) = \emptyset \right.$$

* PROCESI SU IMAJU NEZAVISNIT

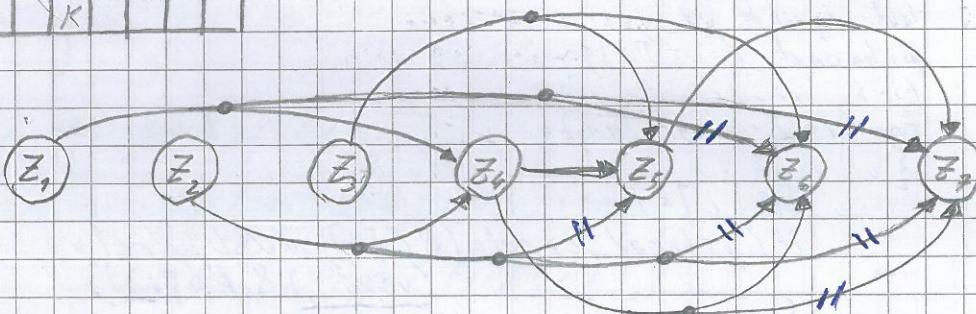
ZAD. 4.2. SUSSTAV ZADATAKA JE ZADAN U OBliku LANCA:

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

ZADATCI IMaju PREDMETE I PREDMETNE U TABLICI:

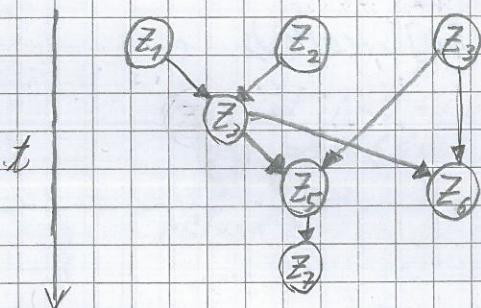
M	1	2	3	4	5	6	7
1	0	0	0	0	K		
2	K		K	0	D		
3		K	D	0	K		
4			K	D			
5				K			

ODGOVITI MAXIMALNO PARALELNU
SUSSTAV ZADATAKA, VZIMAJUCI
U OBZIR NJIHOV MFBUSOBNI
ODNOS U LANCU



II UKLANjanje PREDMETnih ZAVISNOSTI

PARALELNI SUSSTAV ZADATAKA:



► imps. jst u reprezentaciji
najp. ova mjerama

MEDUSOBNO ISKLUJUĆVANJE

Dodatak

dok je (i)

K.O. ; // kritični odsječak
N.K.O. ; // nekritični odsječak

long A;

Dodatak

za $i=1$ do 1000 000

A++;

A = 0.

ispis A; // glazba dodatak

ALGOITOM MEDUSOBNO ISKLUJUĆVANJA

- 4. UVJETI: 0. DODATNO MEDUSOBNO ISKLUJUĆIVO
OBVEZNU KRIJICI ODSJEĆAKA

1. ALGOITOM MORA DODOVATI I
ONDA NADA SU BREZNO IZVODIMA
DODATNI PREDMETI

2. KADA JEONA OD DODATNI ZASTRONE
U N.K.O., TA NE SMJEŠT SPODJEĆI
DAVU DODATNU DA VREB U - K.O.

3. IZBOR NOJA ČE DODATNA UČI U
K.O. MOže JE ODRITI U KOMADNU
VREMENU

// Prvi pokusaj

Dodatak i

dok je (i)

dok je ($Z == 1$); //čekaj

K.O. ;

Z = 0.

N.K.O. ;

// Drugi pokusaj

Dodatak i

dok je (i)

dok je (pravo == j);

K.O. ;

pravo == jj;

N.K.O. ;

PROBLEM: DODATNO ULAZO PREDLOZENO
A U STAVU OBVE PREDSTAVLJU O

PROBLEM: NAO SE VRAĆA U PROBITIVU
DODATNU, ONA MOže ZAPRI
U N.K.O I NO DA PRAVO
ONOZ NEZN MOže

// Treći pokusaj

/* uhesi više zastavica (algoritom
sličan drugom pokusaju, sime
postavi Z prije K.O. */

PROBLEM: IDOMOJUON PREDOM POKUSAJU

// Četvrti pokusaj

/* kao u trećem, sime obrn
redoslijed dok je i postavljen
K.O. */

PROBLEM: ZASNOV

// Peti pokusaj
/* niste */

// Dekterov postupak

```
int PROVO, Z[2]; // bool-ovi - moraju bit postavljeni na 0 ili 1  
dok je(1)  
{  
    Z[i] = 1;  
    dok je (Z[j] != 0)  
    {  
        dok je (PROVO == j)  
        {  
            Z[i] = 0;  
            dok je ((PROVO == j) && (Z[j] == 1)); // radno očekuje  
            Z[i] = 1;  
        }  
    }  
    K.O.; // kritični odsječak  
    PROVO = j; // nije problem da provo prije zavrsnice  
    Z[i] = 0; // jer se nista dogodilo  
    N.K.O.; // nekritični odsječak  
}
```

// Petersonov postupak

```
int PROVO, Z[2];
```

```
dok je(1)
```

```
{  
    Z[i] = 1;  
    PROVO = j; // bio je drugi doje provo drugi dojeti  
    // koj je onda daje pravo uvoj * /  
    dok je ((PROVO == j) && (Z[j] == 1)); // radno očekuje  
    K.O.;  
    Z[i] = 0;  
    N.K.O.;  
}
```

// Lamportov postupak

```
dok je (1) → PROBLEM! => RODOSLJIVO  
{  
    ULAZ[i] = 1;  
    ZAONJI++;  
    BROJ[i] = ZAONJI; } BROJ[i] = A + ZAONJI;  
    ULAZ[i] = 0;  
    za (j=0; j < N; j++)  
}
```

čitaj ULAZ[j];

dok je (ULAZ[j] == 1) čitaj ULAZ[j]

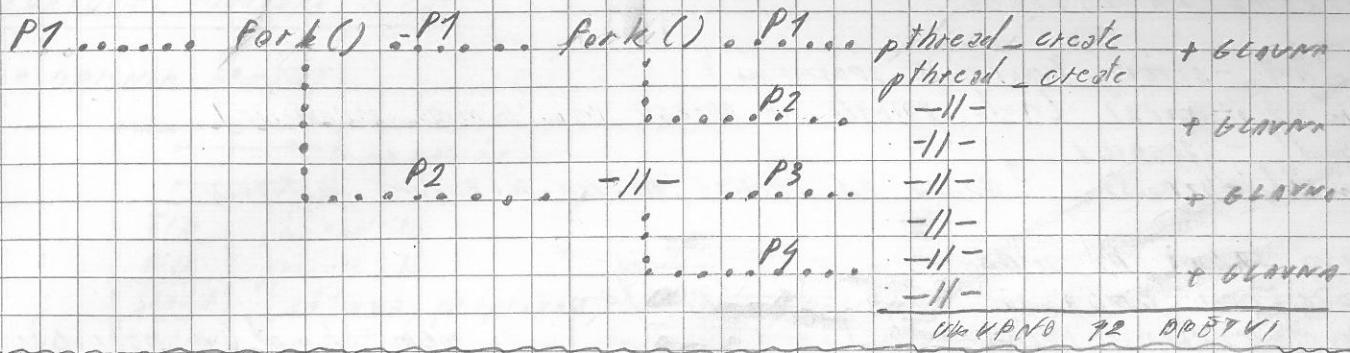
čitaj BROJ[j];

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

{ // znači je isti broj, gledaj T10

čitaj BROJ[j];

} k.o. RPOJ[i] = 0. N.k.o. ; }



* if $(\text{fork} = 0)$

- DAT CQ 0 008705V P2 => TRUB;
- DIT 18 NÖH1 0001 00010001 P7 => FOLSC;

* PROČESI NO MOGU ROMANILJUĆI IZDUSOBNO BUDUĆI SA SLOVENSKE
NAČIN TO RADIĆI => PA DOPOLNUJU NEMASLJENO PONUDA
MOMAČIJI JEDNO DAKLE

- SKLOPOVSKA POMOĆ MEDIJU OBNOVOM IZDELJUJANJEM

doh je (1)

dok je ($\text{TAS}(\text{zastavica})$); // radno iekanje
 // test and set - ispitaj zastavica (0. lig)
 // u modernim
 // i postavi je na 1
 // arbitralnim
 // \Rightarrow dok je 1 - radno iekanje
 // imamo CAS
 // dok je ($\text{cas}(\&\text{zastavica}, 1, 1)$); // radi isto

K.O.-

$$ZOSTAVICA = 0;$$

N.K.O.;

II PRIPRJEMA ZA 2. LABOS

shmget - stvori (vrati ID spomenika)

shmget - povrzi (vrati pointer na memoriju koju je mohao memotors)

shmat - oslobodi

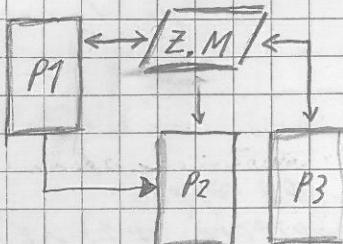
shmatl - izbaci ! NUŽNO ZA SEPVRE, DA NE PONOSTANE MEMOARJE

- prvo stvori, pa onda

povrzi i onda svi

komuniciraju podno

zajednicke memorije

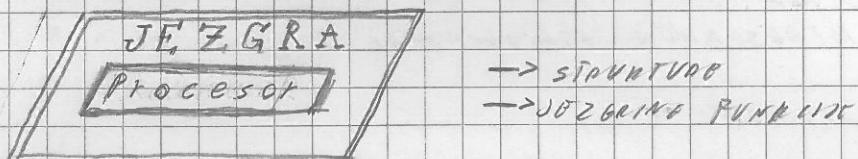


mem = shmat(id, NULL, 0);

→ prvo evog povozivat da
prisuprav slom

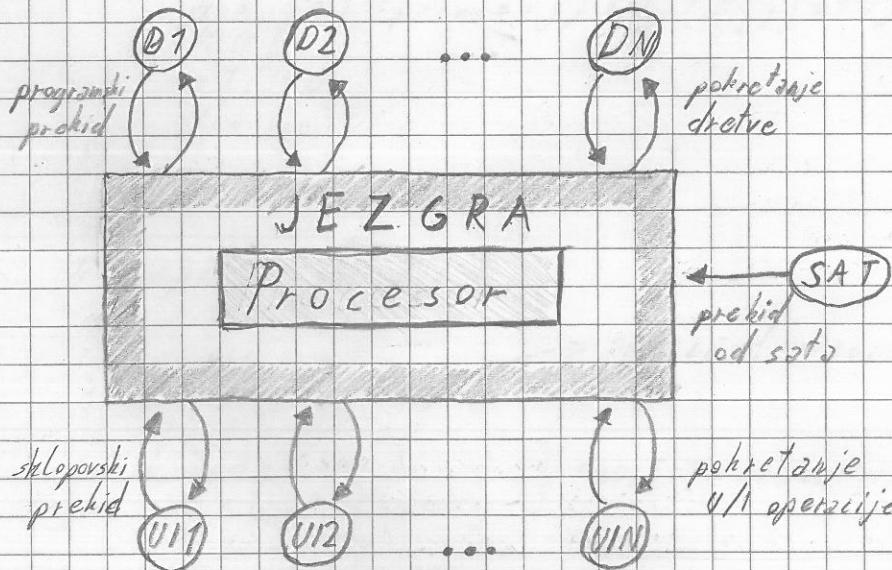
JEZGRA OPERACIJSKOG SUSTAVA

! SKUP JEZGRENIH RUMACIJA I SKUP STRUKTURA PODSETARA JEZGRE



! POZIV JEZGROV ZNAČI POZIV JEZGRENIM RUMACIJIMA,
I NAZIVNI SE ULAZAK U JEZGROU
I ZBIVA SE KAO SE ULAZ U PROSNIĆ

! IZLOZAK IZ JEZGROV SKODI SE NA PONOVLJENJE JEDENO OD ONTOVI
PAJ TOČNU PROCESOR MORA BITI VNOĐEN U MAIN PODE TO DODATN
(najčešći početni main podes)

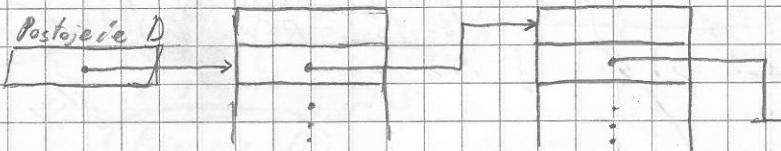


STRUKTURE PODATAKA JEZGRE

• OPISNIK DRŽIVE

•	→	} kazalište.
•	*	
TID	thread ID	
PID	process ID	
STANJE	{ aktivna, pripravna, blokirana, pasivna }	
PRIORITET		
PA	početna adresa memorije (četverochodni adresnog podprostora)	
VBL	veličina memorije (četverochodni adresnog podprostora)	
ADR. T1	adresa prve instrukcije	
KASNIJNJE		
PROSTOR		
ZAPORNJA	} kontekst u vremenom smislu	}
NONTIČKA		
DIREKCIJA		
RS	registar stanja	
PC	program counter	

- DRŽIVE SO NALAZE U LISTAMA



// konstante binarnog semafora

$B.Sem[i].v = 1;$ // početne vrijednosti
dok je (1)

Cekaj - $B.Sem(i);$

K.O.:

Postavi - $B.Sem(i);$

N.K.O.:

}

// ! Ovo se bi trebalo postaviti - SHV.01 i nastavi

// ULAZNA

dok je (n)

{

dohvatiti podatke;

Cekaj - $B.Sem(1);$

pošalji podatke radnoj skotri;

Postavi - $B.Sem(2);$

}

// RADNA

doh je (1)

{

Cekaj - $B.Sem(2);$

primi podatke;

Postavi - $B.Sem(1);$

obradi podatke;

Cekaj - $B.Sem(3);$

pošalji ulaznoj D;

Postavi - $B.Sem(4);$

// IZLIZNA

doh je (1)

{

Cekaj - $B.Sem(4);$

primi podatke;

Postavi - $B.Sem(3);$

prikazi;

}

// POČETNE VRIJEDNOSTI

$B.Sem[1].v = 1$

$B.Sem[2].v = 0$

$B.Sem[3].v$

$B.Sem[4].v$

! NAPOMJENJE:

- NA JEDNOJ PROCESORSKOM RAČUNALO MEDIUSOBNO ISKUŠAVANJE JEZGARNIH PUNKCIJA OBAVLJA SE ZABRINOM PROTEKCIJE
- NA VIŠEPROCESORSKOM RAČUNALU NE POSTOJI MEDIUSOBNI ZABRANJIVANJE PA TEKIDANJA NA DRUGOM PROCESORU
- ZASTOVICA OBRADNI JEZGRI
- 3 VESTI TEKIDINA: PROGRAMSKI, SNOOPING, PREHOD NA SADA

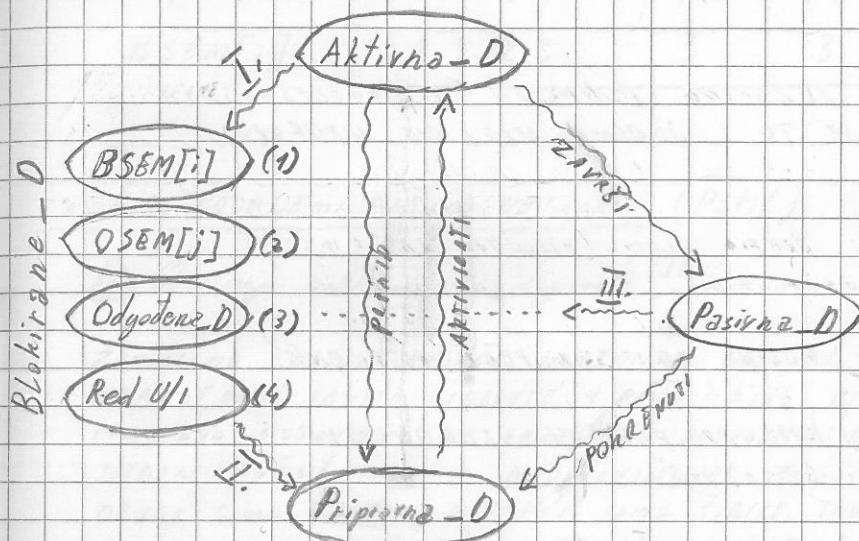
! OPCI SEMAFOR

- JEZGARNI MELINIČNIH MOGI SLUŽI ZA SINKRONIZACIJU DRUGIH
I SASTOJI SE OD STRAVOŠU PODATAKA I JEZGARNIH PUNAČIJA
- SASTOJANO JE ISTA, ALC VARIOVANOSTI, SU DVE:

 - $[-\infty, 0]$ NEPOGOZNO }
 $[1, +\infty)$ POGOZNO } OS - DAKSTOIN OPCI SEMAFORA

- OSM - BROJNIKI SEMAFORA
 - KAD SE NAJDE OPCI SEMAFOR MIŠCI SE NA OVIS
 - VARIOVANOSTI: 0 NEPOGOZNE
 $[1, \text{MAX } V]$ POGOZNE

MOGUĆA STANJA DIREKTE



- I.
- Aktivna \rightarrow (1): ČEKATI BSEM
 - Aktivna \rightarrow (2): ČEKATI OSSEM
 - Aktivna \rightarrow (3): ZAKASNITI SLOB (SLEEP)
 - Aktivna \rightarrow (4): ZAPRIETI U/I
- II.
- (1) \rightarrow Pripravna: POSAVITI BSEM
 - (2) \rightarrow Pripravna: POSAVITI OSSEM
 - (3) \rightarrow Pripravna: OTKUCATI SLOB
 - (4) \rightarrow Pripravna: PREKID U/I NAPRAVIT
- III.
- Pripravna \rightarrow (3): ZAKASNITI

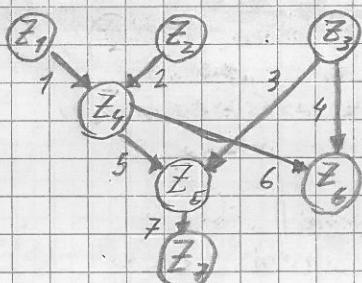
SEM	ZABRANA	CRVENO	
		NEMA NIROVA	IMA NIROVA U RAZ
Cekaj-BSEM	$V=0;$ PROTI SEMAFOR;	PRIJEDI U RAZ BSEM;	
Postavi-BSEM	$V=1;$ ili NIŠTA		POSTAVI PARI OPISNA IZ RAZA BSEM U RAZ Pripravne-D;
Cekaj-OSSEM	$V=-1;$ PROTI SEMAFOR	PRIJEDI U RAZ OSSEM;	
Postavi-OSSEM	$V++;$		POSTAVI PARI OPISNA IZ RAZA OSSEM U RAZ Pripravne-D;

IZLAZAK 12 JEGEZBO

- * SVODI SE NA ANTIVIRIONE NERE DRETVE,
NAJJESENJIJE I. PAIPANJE (NO MORA)
- * AKTIVIRATI IZ PODE Praprema - Di:
 1. PRIMJESTIOI Povi OPSNIH IZ ADO Praprema-D u ADO Platno-D
 2. OGMOVITI MONTESTI 12. OPISNIH Aktivna-D
 3. VEDRITI SO U PODANJU DROVU
- 3.7. OBNOVITI RS
 - 3.7.1. VAOVITI SO U NOVINI PRED PODANJU DROVU
 - 3.7.2. OMOGUĆITI PRIMJESTJE, NO JE BILA EMALIJNA
 - 3.7.3. ANTIVIRIONI ODLOVAKOMA KAZNUJKU STOGA
TO ADOSSIRATI ODGOVARAJUĆI ADOSEN PREDSTO
- 3.2. OBNOVITI PC

! RADNO ŽEKONJE SB PBI ULASAK U JEGEZBINO PUNKTO VJEĆPREČOŠOŠKOG SUSTAVA NE MOže IZBJEĆI, ALI JE TO I JEĐINA NIEGOVA UPORABA

ZAD. 5.7. ZA OSVARENJE SUSTAVA DRETVI PREMA SLICI (RAZJEDNO ZAD. 4.7.)
koriste se binarni / opći SEMAFORI



- a) kolika je semafora potrebno?
- b) koje su pojavne varijabne semafora
- c) neka je T_i vlast zadataka i.
prosječni svaki zadatak T_i i T_j
s minimiznim brojem procedura

a) BSFM - onako koliko ima stranicu

$$\#BSFM = 7$$

OSFM - onako koliko ih ima slikece pojma strani

$$\#OSFM = 4$$

b) $BSFM[i] = 0$

$OSFM[i] = 0$

Svi semaproksi su neprolozni

c) $BSFM: T_2': T_2;$

Postavi - $BSFM(2);$

$T_2': Cekaj - BSFM(1);$

$Cekaj - BSFM(2);$

$T_4:$

Postavi - $BSFM(5);$

Postavi - $BSFM(6);$

$OSFM: T_2': T_2;$

Postavi - $OSFM(1);$

$T_2': Cekaj - OSFM(1);$

$Cekaj - OSFM(1);$

$T_4:$

Postavi - $OSFM(2);$

Postavi - $OSFM(3);$

ZAD.5.4. STRUKTURE PODATAKA DRETVO

U PROMATRANOM DRETVOM STANJE SUSTAVA JE SLJEDOĆE:

- DRETVO 1 JE AKTIVNA

- DRETVO 2 I 3 SU U REDU BINAONOG SEMOPADA

- DRETVO 4, 5 SU U REDU PRIPRAVNIIH DRETVI

AKO TADA AKTIVNA DRETVO (1) POZOVU JE ZGODNO PUNJENI Postavi - BSOM(2)

PAKO ČE IZGLJEDET STAVAKA PODATOK JE ZBEG NACION POZIVU?

SVI REDOVI SU ORGANIZIRANI PO KROU PRISPLODNI. => FIFO

	PRIM	MEGUVEZATAT	RJEŠENJE
Aktivna - D	1	-	-
Pripravka - D	4,5	4,5,7	9,5,1,2
BSOM[i]	2,3	2,3	3

ZAD.5.2. PROBLEM PUŠAĆA CIGARETE (Patil)

⚠ Ovo nije reklama za cigarete!!!

ZAMISLIMO SUSTAV PUŠAĆA I JEDNOM DRETVOM TRGOVCEM. SVAKI PUŠAĆ
NEPOSTANO SAVLJO CIGARETE I PUŠI. KAO BI SE SAVLA I POPUŠILA
CIGARETA POTREBNO JE IMATI TAKI SASTOJAKA: DUHAN, PAPIR I ŠIBICE.
JEDAN PUŠAĆ IMA U NEORGANICNIM KOMponentama SAMO PAPIR,
DRUGI SAMO DUHAN, A TREĆI SAMO ŠIBICE. TRGOVAC IMA SVA TRI
SASTOJKA U NEORGANICNIM KOMponentama. TRGOVAC NASUMILJE STAVLJA
NA STOL DVA RAZLIČITA SASTOJAKA. PUŠAĆ KOJIIMA JE ODOŠAO
SIGARICIRAN TRGOVCU, SAVLJA CIGARETU I PUŠI. TRGOVAC STAVLJA NOVA
OM SASTOJAKA NOVA DVA SASTOJAKA NA STOL I CIKLUS SEZ PONAVLJA.
NO POČETNU STOL JE POZAN. NAPISATI DRETVO PUŠAĆA I TRGOVCA
TAKO DA SE ONE MOGUĆOBNO ISPRAVNO SINKRONIZIRAJU POMERI
BINARNIH SEMOFORA.

dretva Trgovac()

{ dok je (1)

Cekaj - BSOM (stol - pozvan);

Stavi - dva - na - stol;

Postavi - BSOM (ono što je stavljen);

{ dretva Pušac [ono što tragi] ()

{ dok je (1)

Cekaj - BSOM (ono što tragi);

Vazi - sastojke;

Postavi - BSOM (stol - pozvan);

Smotrij - cigaretu - i - prsi);

//Početne vrijednosti

BSOM (stol - pozvan).v=1;

BSOM (duhan - papir).v=0;

BSOM (papir - šibice).v=0;

BSOM (šibice - duhan).v=0;

/*

* Moras pisat svaku dretvu zasebno

* (i u onom postavljanju kod Trgovca

* su tri linije bilo)

*/

ZAD.5.5. VRUČUJCI

MODELIRATI VRUČUJCI S DVA TIPO DODAVCI:

- DODAVCA POSJETIOČELJU (POŠTOTOČELJU NOJI ŽELO ČA VJEĆA NA VOZIĆU)
- JEONOM DODAVCOM VRUČUJCI

DODAVCA POSJETIOČELJU JE NO SMJE DOZVOLITI UVEĆATI NA VOZUĆJAK
KAO KOMO VIŠE MESTO (KOMISITI POMOĆNU VRIJEDNOST BR-MJESTA)
A POKRENUTI VRUČUJCI TEK KAO JE PUN.
KOMISITI OPĆI SEMAFOR I DODAVNO VRIJEDNOSTI

dretva Posjetitobj()

```
{
    Cekaj_OSEM(VRULJAK);
    Sjednici;
    Postavi_OSEM(SJEO);
    Cekaj_OSEM(KRAS);
    Ustani_i_sidi;
    Postavi_OSEM(SISAO);
}
```

dretva Vruljak()

```
{
    dok je (1)
```

```

    za (i=1 do BR_MJESTA)
        Postavi_OSEM(VRULJAK);
    za (i=1 do BR_MJESTA)
        Cekaj_OSEM(SJEO);
        Pokreni_Vruljak();
        Zavrsni_Vruljak();
    za (i=1 do BR_MJESTA)
        Postavi_OSEM(KRAS);
    za (i=1 do BR_MJESTA)
        Cekaj_OSEM(SISAO);
```

// Početne vrijednosti:

```

    OSEM(VRULJAK).v = 0;
    OSEM(SJEO).v = 0;
    OSEM(KRAS).v = 0;
    OSEM(SISAO).v = 0;
```

/* u praksi ne bude vraka
sve 0 */

ZAD.5.3 PRIORITETNI RADNOVI I SEMAFORI

U JEONOPREDČEŠĆOM RADNOMU POKRENUT JE SUSTAV DRETVI D_1, D_2, D_3, D_4
S PRIORITETIMA 1, 2, 3 i 4 RES PREDSTAVLJU. MAJ VISE PRIORITET JE 4. SVI ZADANI
KOJI DRETVE OBILJUJU SU D_x . RES PRIORITETNI DODAVCI I PRED SEMAFORI SO
PRIORITETNI. AKTIVNA JE DRETVA KOJA JE PRAVA U REDU PRIPRAVNA (NEMA I POSLEDNJE
REDA AKTIVNIH DRETVI). PRIJE POKRENJANJA SUSTAVNA DRETVI SE SEMAFORI S JU BLO
ZAVRŠETOM. NAPON NEKOG VREMENJA SVE DRETVE SO NAPU U REDU SEMAFORI. SA
AKO SE TADA POZOVU PREDLOŽINA Postavi_BSOM(S), SLEĆI SE ISPISETI?

// ZADANO

Dretva D_x

```
{
    dok je (1)
```

```

    Cekaj_BSOM(S);
    pisi(Px);
    Postavi_BSOM(S);
    pisi(Zx);
```

Red BSOM v Pripravne-D izlaz

4 3 2 1	0	1	
3 2 1	0	42	P4
2 1	0	46 32	Z4
4 2 1	0	32	P3
2 1	0	48 36	P4
1	0	46 36 22	Z4
4 1	0	36 22	Z3
4 3 1	0	22	P2
3 1	0	42 26	P9
1	0	46 32 26	Z4
4 1	0	32 26	P3
1	0	42 36 26	P4
-	0	46 36 26 72	Z4
4	0	36 26 72	Z3
4 3	0	94 72	Z7

4 3 2	0	12	P1
3 2	0	16	P4
2	0	16	Z9
4 2	0	16	P3
2	0	16	P4
7	0	16	Z4
5	0	16	Z3
4 3	0	16	P2
3	0	16	P4
-	0	16	Z4
4	0	16	P3
-	0	16	P4
-	1	16	Z4
-	0	16	P4
:	:	11	:

* varijanta rezultata:
- unutar dok je (1) $\Rightarrow z_2 \neq 0$ do 8

MEDUDRSTVENA KOMUNIKACIJA I KONCEPCIJA MONITORA

! Neviđeni ore kombinacije potrebnih operacija sa X provođenjem i Y galvanizacijom
 \Rightarrow maksimalno je 2 binarna i 2 opća

II za L1BOS 32

Dretva 1

{ :

-ipem

-iperm -im -memonija
-s -semantika

(4)

ZASTAVI:

Cekaj - BSEM(1); \leftarrow OSIM ta
Cekaj - BSEM(2); \leftarrow postime
k.o.

Postavi - BSEM(1);
Postavi - BSEM(2);

{ :

Dretva 2

{ :

{ (2)

Cekaj - BSEM(2);

Cekaj - BSEM(1); \leftarrow (3) ZASTAVI!
k.o.

Postavi - BSEM(1);
Postavi - BSEM(2);

{ :

ZAD

SINKRONIZACIJI DRETVOV VODIĆA I PISMAKO SVARUJU MOLEKULU VODE.

Dretva atom - H()

{

Postavi - OSIM (STIGAO - H);
Cekaj - OSIM (SPAJANJE);
pričekati se - atomu - kisik;

}

II granice vrijednosti

BSEM (k.o). v = 1 /stiti, k.o.
OSIM (STIGAO - H). v = 0
OSIM (SPAJANJE). v = 0

Dretva atom - O()

{

Cekaj - BSEM (k.o);
Cekaj - OSIM (STIGAO - H);
Cekaj - OSIM (STIGAO - H);
Postavi - OSIM (SPAJANJE);
Postavi - OSIM (SPAJANJE);
spajanje - atom - u molekulu - vode;
Postavi - BSEM (k.o);

}

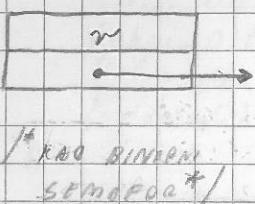
! 3. NUJNA UVJETA ZA MOGUĆIJE POKLONOG ZASEOJA

- NEKO SPOSOŠTVO U ISTOM ČASU MOŽE MOGUDIĆI SAMO JEĐANA PRETRVA
- DRĘVI SO SPOSOŠTVO NE MOŽE ODRZET
- DOSTVA DRJI DOVOLJENO JOS SPOSOŠTVO DA ČINI NA
DOVOLJNU DRUGOG SPOSOŠTVA

MONITOR

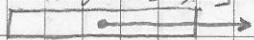
- JEZGRIN METONIZAM KOJI SLUŽI ZA SINCHRONIZACIJU DOSTVI
KOJI SE SASTOJI OD STRUKTURA PODLOGA I JEZGRINIH FUNKCIJA

Monitorski
semaphore (M)



Redovi uvjeta

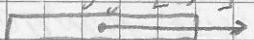
Red-uvjet [M, 1]



Funkcije za ostvarenje
monitora

→ jezgrina funkcije : 5

Red-uvjet [M, 2]



→ monitorska funkcija
/* ovo sam gaoio */

JEZGRINE FUNKCIJE

1. Uti_v_monitor(m);
2. Izadi_iz_monitora(m);
 - Uysti_v_red_uvjeta (M, k);
 - 3. Cekaj_v_red_uvjeta (M, k);
 - 4. Osloboidi_iz_reda_uvjeta(k);
 - 5. Osloboidi_sve_iz_reda_uvjeta(k);

- // mutex_lock (& M);
// mutex_unlock (& M);
// NO MONITRI SE IZPRAVNO
// cond_wait (& k, & M);
// cond_signal (& k);
// cond_broadcast (& k);

⇒ ZAŠTO?

- PO KODI BIDI U SERVIRU
KAO NO MONITR SVE

! PRAVILA KOJIM SE TREBA DRŽATI PRED KADOM IZGRADNJE MONITORA

1. MONITORSKE FUNKCIJE JE ZAŠTEJENA MONITORSKIM SEMAFOROM
⇒ PISI UAT I 12001 PAJE INFOR. MJH
2. MONITORSKE FUNKCIJE PAJE KRETJICOG ODSJEKA ISPIJEDNO UVJET
I AKO UVJET NIJE 2000 VOLJEN SVASTOM DOSTVU U ACO UVJET
⇒ ČEKAJ_V_RED_UVJETA
3. MONITORSKE FUNKCIJE POSLJE KRETJICOG ODSJEKA PA POGRESI
OSLOBOIDI DOSTVU IZ PREDNOVUVJETA
4. ISPIJEDNA UVJETA JE NE OBAVLJA S AKE JE (IF), NOGO
S dok je (while)
⇒ ACO JE POGRESA OSLOBOIDI MON PONOVNO ISPIJED UVJET

NAPOMENA ZA 3. LAB

- ZADATAK 3b SE RADI S OPERACIJAMA (MONITRI Nisu VIMNIJENI ZA PROCES)

- ACO JE NOKA DOSTVA U MONITRU MIJEGUNA DRUGA MO NESTI UJIS,
ESIMA AHO OVA VOĆ IONAKO IZLAZI

$$\begin{aligned} * \text{cond_wait} &= 1, 12001 \text{ IZ MONITORA } 1 \text{ UBI U 200 UVJETU} \\ &= 2, 12001 \text{ IZ RJEŠENI UVJETA } 1 \text{ UBI U MONITORA} \end{aligned}$$

II PROBLEM PET PILOZOPI

dodata Fibozop (n)

{ dok je (1)

misli;

m - oti - u - ko();

jedij;

m - izact - iz - ko();

}

m - oti - u - ko()

{

mutex_lock(&M);

while (bijevi ili desni stapici zavzet)

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

uzmi_stapice();

mutex_unlock(&M);

}

// struktura

monitor M;

5 redova ujete red[i];

m - izadi - iz - ko()

{

mutex_lock(&M);

spresti_stapice();

if (bijevi susjed imao oba stапice)

cond_signal(&red[bijevi susjed]);

if (desni susjed imao oba stапice)

cond_signal(&red[desni susjed]);

mutex_unlock(&M);

}

ZAD. 6.7 STARI MOST

STARI MOST JE OSIJI MOST I STOGA PREDSTAVLJA OBONIJENJA ZA POMER.
NA NJEMU ISPODOMO SMJUĆE BITI NAVJEŠTE JU AUTOMOBILIMA KOJI VOZDE
U ISTOM SMJERNU. SIMULIRAJ AUTOMOBILU POKRIVOM AUTO KOJI OBVEŠTA NIZIC
NAVEĐOVNO RAVNJU: NAPISATI PREDODGOVOR MONICOOSNIH FUNKCIJA
Popri-se-na-most (smjer), Sidi-s-most();

dodata Auto (smjer) // smjer 0 ili 1

{ Popri-se-na-most (smjer);

projedi-most; // K.O.

} Sidi-s-most();

ZAB.6.2. MICROSOFT OVI I LINUX PROGRAMERI

U ISTOJ ZBENOVI RADOV MS I LINUX PROGRAMERI. ZGRADOO IMA JEDAN RESTORAN, KOJOGA PROGRAMOMI MORAJU DISTRIBUITI. U ISTOM TRAVNIKU U CESTOVACU SNIJU BICIKL SEME JEDNOU VASEM PROGRAMATORA (ILI JE RESTORAN PRAZAN) SVAKI PROGRAMERIMA SE SVIJEĆI OBCLIN:

programer (vrsta)

```
{  
    ucti (vrsta);  
    obzri;  
    izadi (vrsta);  
}
```

A) RIJEŠITI ZADATAK MONITOARIM.

B) RIJEŠITI ZADATAK BEZ POJAVE JEBOVANJA, TAKO DA NEMISE N PROGONJAN JEDNU VASE, A YU V RESTORAN, AKO JE N PROGOMJANA DUGO VASE, NAIZMENICE IMA N VASNIH PA N DAVBIN.

ucti (vrsta)

```
{  
    mutex_lock (&M);  
    ceka [vrsta]++;  
    dok je (br [i-vrsta] > 0) || ((SITI == N) && (ceka [i-vrsta] > 0))  
        cond_wait (&red [vrsta], &M);  
    br [vrsta]++;  
    ceka [vrsta]--;  
    SITI++;  
    mutex_unlock (&M);  
}
```

izadi (vrsta)

```
{  
    mutex_lock (&M);  
    br [vrsta]--;  
    ako je (br [vrsta] == 0) {  
        SITI = 0;  
        cond_broadcast (&red [i-vrsta]);  
    }  
    mutex_unlock (&M);  
}
```

// STRUKTURE PODATKA

```
vrsta // 0 (MS) ili 1 (Linux)  
int cekac [2] // brojac ispred restorana  
int br [2] // brojac u restoranu  
red [2] // redovi vrjeda  
int SITI = 0
```

ZAD. 6.3. PING - PONG DRŽVU

Simulirati dva drživa ping i pong. Drživo ping u bilošinskom položaju ispisuje "PING", a drživo pong "PONG". Simulirati da na izmjenično ispisuju "PING" "PONG": A) somoporimno B) monitorom

A)

drživo Ping

```
{ while(1){  
    Cekaj -> BSOM(1);  
    Ispisi ("Ping");  
    Postavi -> BSOM(2);  
}
```

B)

drživo Pong

```
{ while(1){  
    Cekaj -> BSOM(2);  
    Ispisi ("Pong");  
    Postavi -> BSOM(1);  
}
```

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

B)

drživo Ping

```
{ while(1){  
    mutex_lock(&M);  
    dok_je (zaDnja == PING)  
        cond_wait (&red[PING], &M);  
    zaDnja = PING;  
    Ispisi ("Ping");  
    cond_signal (&red[PONG]);  
    mutex_unlock(&M);  
}
```

drživo Pong

```
{ while(1){  
    mutex_lock(&M);  
    dok_je (zaDnja == PONG)  
        cond_wait (&red[PONG], &M);  
    zaDnja = PONG;  
    Ispisi ("Pong");  
    cond_signal (&red[PING]);  
    mutex_unlock(&M);  
}
```

Neovrhovanje: zaDnja = {0(PING) ili 1(PONG)}, monitor M, red[2]

ZAD. 6.4. PARKING

Novi parking (npr. park-ing) ima dvije stanice: jednu za ulaz i drugu za izlaz. Automobili ulaze na parking u pomoci daljinskog uređaja. Svaki vrednost im je jedinstveni broj za koji u bozi sustava postoji definisana stanica: 0 - auto nije na parkingu; 1 - auto je na parkingu. Ulagaj se automobilima mogući samo u stanju 0, a izlazak samo u stanju 1. Automobil ne može ući ako je na parkingu veći MAX automobila. Simulirati sistem parkirana, t.j. neispoljivu ulaz(id) i izlaz(id) koja pozivaju auto s brojem uređaja id. Ponovo se gleda da je evidentna vezanost između automobila te se ne promjeni stanje uređaja i broja mesta. Ako je poprečno, podaci imaju parstvu bozi i može promijeniti stanje automobila i broj mesta. Dodati moraće se poneti porti-pusti-auto-kaz(id) i porti-pusti-auto-na-parking(id) koje pozive porti puto pozivajući odgovarajuće stanje uređaja i broj mesta.

ulaz(id)

```
{ mutex_lock(&M);  
while (br_mjesto == 0 || auto[id] == 1)  
    cond_wait (&ulaz, &M);  
auto[id] = 1;  
br_mjesto++;  
mutex_unlock(&M);  
}
```

izlaz(id)

```
{ mutex_lock(&M);  
while (auto[id] == 0)  
    cond_wait (&izlaz, &M);  
auto[id] = 0;  
br_mjesto--;  
cond_signal (&ulaz);  
mutex_unlock(&M);  
}
```

portit - posti - auto - van (id)

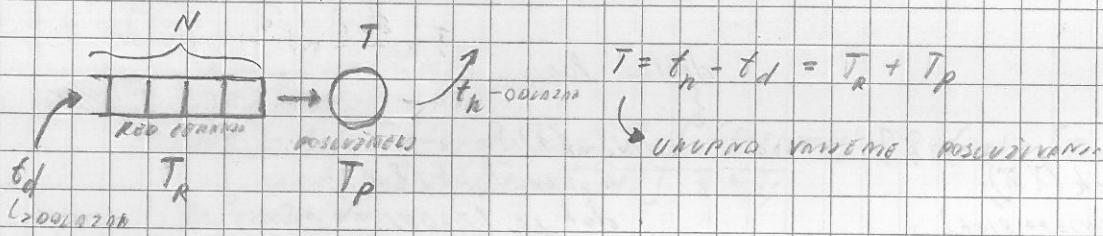
{
motorx_lock (&M);
auto[id] = 1;
br_mjesta --;
cond_signal (&izlaz);
motorx_unlock (&M);
}

portit - posti - auto - na - parking (id)

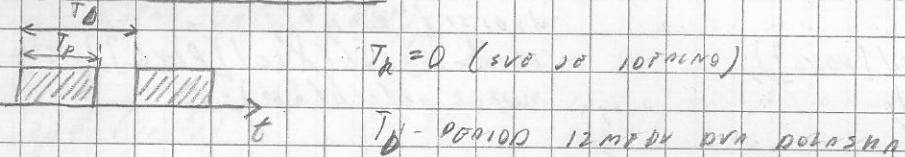
{
motorx_lock (&M);
auto[id] = 0;
br_mjesta++;
cond_signal (&ulaz);
motorx_unlock (&M);
}

korisnik M; red vrste ulaz, izlaz; int br_mjesta = MAX; auto[id] = 0; int auto[max]

ANALIZA VREMENSKIH SVOJSTAVA RAČUNALNOG SUSTAVA



• DETERMINISTIČKI SUSTAV - SVI DOGLADNI SU POZNATI I PREDVIĐAJUĆI



$$\varrho = \frac{T_p}{T_0} \quad \begin{matrix} \text{ISHODNOST} \\ \text{POSLVJETOĆA} \end{matrix}$$

$$(1) \quad d = \frac{1}{T_0} \quad \begin{matrix} \text{PREHVĀĆENJA DOGLADNA} \\ = BROJ DOGLADNIH POSLOVA U SREDINI SUSTAVU \end{matrix}$$

$$\varrho = \frac{d}{\beta} \quad \begin{matrix} \text{SPOSOBNOST POSLJEDICE DA ODAVI} \\ \beta \text{ POSLOVA U JEDINICI VREMENA} \end{matrix}$$

* U DETERMINISTIČKOM SUSTAVU JE DOZVOLJENO $\varrho = 1$

! U NEDETMINISTIČKOM SUSTAVU $\varrho > 1$ KRNJEVA PREGIJANA PRED

(1) $\frac{1}{d}$ PROSJEČNO VRIJEME IZMEĐU !
 d DOGLADNI DVA POSLOVA U SUSTAVU !

(2) $\frac{1}{\beta}$ PROSJEČNO VRIJEME IZMEĐU !
 β BROJ OBLOGA

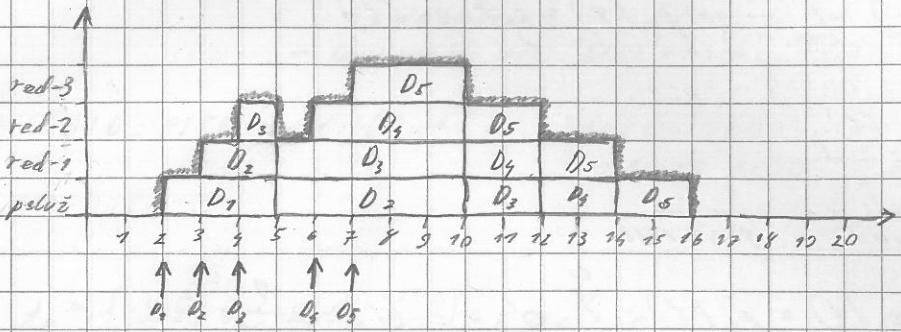
T - PROSJEČNO TRAJANJE ZADOVLAŽAVANJA POSLA U SUSTAVU !

n - PROSJEČNI BROJ POSLOVA U SUSTAVU !

* DETERMINISTIČKI SUSTAV MOŽE BITI I PERIODIČNI \Rightarrow POSLOVI SE PONAVLJAVU

ZAD. 7.1 PROSTAVIMO DA SUŠOVI OBRAZUJE 5 POSLOVA KONTA U NIGDJE DOGLAZD PONIČENO
S PERIODOM OD 20 JEDINICA VREMENINA. DO TOLUKI DOGLAZDANA U PAVOS PERIODI
ZAPORUJE S $t=0$, FRAZANTE POSLOVA NAVODJENI SU U TABLICI

	D_1	D_2	D_3	D_4	D_5
T_1	2	3	4	6	7
T_2	3	5	2	2	2



$$\Rightarrow \text{POVARI} = 35$$

$$\text{ali } T_1 + T_2 + T_3 + T_4 + T_5 = 5 \cdot 7 = 35$$

$$\bar{T} = \frac{35}{20} = 1,75$$

$$\bar{h} = \frac{5 \cdot 7}{20} = \frac{35}{20} = \frac{7}{4}$$

$$\boxed{\bar{h} = d \cdot \bar{T} / \text{Littleovo pravilo}}$$

* DOGLASI \Rightarrow POISSONOV RAZDODIR

* DOGLASI (ODNOSNO OBRAZ) \Rightarrow EKSPONENCIJALNA RAZDODIR

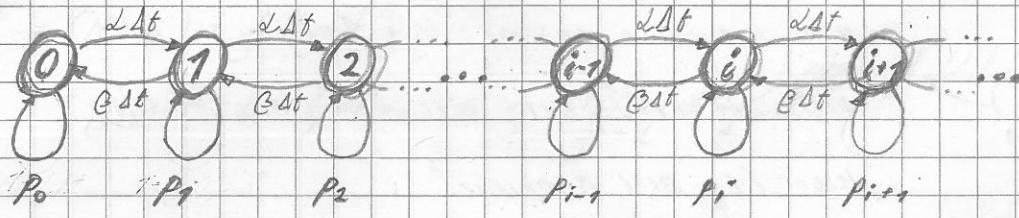
► POGLEDATI MATEMATIKU IZVOĐENJA SVAKOČIM STVARI AKO JE ZANIMA

$$\text{ULAZ: } p(k(\Delta t) = 0, d) = e^{-d\Delta t} \approx 1 - d\Delta t$$

$$\text{IZLIZ: } p(k(\Delta t) = 0, \beta) = e^{-\beta\Delta t} \approx 1 - \beta\Delta t$$

VJESENOST NA NORMA NIRO

1.) MARKOVOV LANAČ



$$p_0 = 1 - \alpha \Delta t$$

$$p_1 = (1 - \alpha \Delta t)(1 - \beta \Delta t)$$

$$= 1 - (\alpha + \beta)\Delta t - \frac{\alpha \beta \Delta t^2}{2}$$

$$\approx 1 - (\alpha + \beta)\Delta t = p_i$$

2.) "PROVJETI" MARKOVOV LANAČ

$$p_0(t + \Delta t) = (1 - \alpha \Delta t)p_0(t) + \beta \Delta t p_1(t)$$

$$p_1(t + \Delta t) = [1 - (\alpha + \beta)\Delta t] p_1(t) + \alpha \Delta t p_0(t) + \beta \Delta t p_{i+1}(t)$$

3.) MARKOVOV LANAČ A PROSTAVITI \Rightarrow PIPF. JEDNODIBICE \Rightarrow NOVI MEDIJUMOVI

$$\frac{dp_i(t)}{\Delta t} = 0 = p_i(t + \Delta t) - p_i(t) \quad \left| \begin{array}{l} \Delta t \\ \Delta t \rightarrow 0 \end{array} \right.$$

$$0 = -(\alpha + \beta) \cdot p_i(t) + \alpha p_{i-1}(t) + \beta p_{i+1}(t) \Rightarrow p_{i+1}(t) = (1 + \frac{\alpha}{\beta})p_i(t) - \frac{\alpha}{\beta} p_{i-1}(t)$$

$$0 = -\alpha p_0(t) + \beta p_1(t) \Rightarrow p_1(t) = \frac{\alpha}{\beta} p_0(t)$$

$$p_1(t) = g \cdot p_0(t)$$

$$p_{i+1}(t) = (1-g) \cdot p_i(t) - g \cdot p_{i-1}$$

$$\Rightarrow p_2(t) = g \cdot p_0(t)$$

$$p_2(t) = (1-g) \cdot p_1(t) - g \cdot p_0(t) = g^2 \cdot p_0(t)$$

$$p_3(t) = (1-g) \cdot p_2(t) - g \cdot p_1(t) = g^3 \cdot p_0(t)$$

⋮

$$\underline{p_i(t) = g^i \cdot p_0(t)}$$

4.) 1200000 v god => upitivanje o funkciji

$$\sum_{i=0}^{\infty} p_i(t) = \sum_{i=0}^{\infty} g^i \cdot p_0(t) = p_0(t) \cdot \sum_{i=0}^{\infty} g^i = p_0(t) \cdot \frac{1}{1-g}$$

$$\sum_{i=0}^{\infty} p_i(t) = 1 \text{ (zbog svih vjerojatnosti)} \Rightarrow p_0(t) = 1-g$$

$$\underline{p_i(t) = (1-g)g^i}$$

5.) $\bar{n} = ? \quad \bar{T} = ?$

$$\bar{n} = \sum_{i=0}^{\infty} i \cdot p_i(t) = (1-g) \sum_{i=0}^{\infty} i \cdot g^i = (1-g) \cdot \frac{g}{(1-g)^2} = \frac{g}{1-g}$$

$$\boxed{\bar{n} = \frac{g}{1-g}} = \frac{d}{B-d} = d \cdot \frac{1}{B-d} \Rightarrow \boxed{\bar{T} = \frac{1}{B-d}}$$

PR. 7.2

BROJ POSLOVA O 30 SLOVU: $M = 17N$

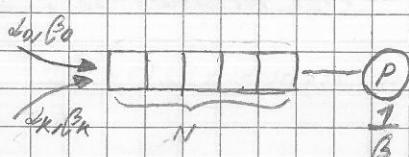


PROIZVODI SE IZ BITI BLOKIMA, AKO: $i \leq N$

KAKO DA BITI BLOKIMA?

$$\begin{aligned} p(i \geq N) &= 1 - p(i \leq N) = \sum_{i=0}^{\infty} p_i(t) - \sum_{i=0}^N p_i(t) = \sum_{i=N+1}^{\infty} (1-g)g^i = g^N(1-g) \sum_{i=1}^{\infty} g^i \\ &= g^N \cdot (1-g) \cdot \frac{g}{1-g} \\ &= g^{N+1} \end{aligned}$$

! ORA JE BITNO!



$$\begin{aligned} \bar{d} &= d_k + d_p \quad \left\{ \frac{1}{B} = \frac{d_k}{d_k + d_p} \cdot \frac{1}{C_k} + \frac{d_p}{d_k + d_p} \cdot \frac{1}{C_p} \right. \\ g &= g_k + g_p \quad \left. \right\} \end{aligned}$$

K-RAZSKI POSLOVI

O-DRGI POSLOVI

\Rightarrow IZ OVIH VJEZDOROBO RAJUNAŠ PROSTOK

NAČINI RASPOREĐIVANJA

7. PIPD (PO RDO PAISAJE)

2. KRUŽNO POSLUDJIVANJE (ROUND ROBIN - RR) => ODEGOVIMO KROVNU VREDNOSTU (T_g)
KOLIKO TRAJE OPERATOR,
AKO JE NODOVOLJEN
BIDI GA ODEGO U RED

3. PO PREDSTAVIMA - U pomoći se ovdje dijelite

KOMBINASI S DLM

- U PRAVILU: NJEDI BROJ \Rightarrow VECI, PREDSTAVIT

* SCHED-FIFO : SCHED-RR ; SCHED-OTHER

=> RUND ROBIN S POSITION VERMINT

$$\underline{\text{200.}} \quad t_0 = \{3.5; 0; 2.5; 6.5\} \quad (A, B, C, D) \\ t_A = \{5, 5, 3, 2\}$$

$\text{CH}_3\text{COO}^+ + \text{H}_2\text{O} \rightleftharpoons \text{CH}_3\text{COOH}$ (acid dissociation)

$$\tan(A) = 2, \tan(B) = \tan(C) = \tan(0) = 1 \quad (\text{важно!})$$

"GAKOBNA RODÔNICA"

Z 10.7.9. ZAHODJENI ZA OBRAOV PODLE JEZU POISSONOVOS RAZDORU s d = 2 i, A VLASTNIM OBRAOEM IMI FUNKCIONALNEM RAZDORU T = 0,5 n

$$P(N \geq 5) = S^5 = \left(\frac{\alpha}{\beta}\right)^5$$

) uzmis kalkulator i nesiuvinis

$$\bar{T} = \frac{1}{\beta - \alpha} \Rightarrow \beta = \frac{1}{\bar{T}} + \alpha \Rightarrow S = \frac{\alpha}{\frac{1}{\bar{T}} + \alpha} = \frac{1}{2}$$

ZAD.3. ZA NEKI WEB SUSTAV S JEDNIM POSLUŽILOM PROŠUJĆEM BAO ZAHTJEVA U
MINUTI JO 700, ODEJ JO SNAGA POSLUŽILOA ZNAJNO VOĆA, ON IH MOŽE
OBRODITI 300 U MINUTI (PROSJIČNO). KOLIKI JE NAJVĆI POKLONAK POSLOVAK POSLUŽITELJSKOG
VRIMONA MOŽE PREDVIRATI ZA DRUGO USLUGU, A DA KLIENCI I DALJO
NE ČEKAJU VIŠE OD DVE DNEVE SOKUNOF NA SKOĆ ZAHTEVA (PROSJIČNO)?

$$d = 100/60 \quad \ddot{\sigma}^2 = 5/3 \dot{\sigma}^2$$

$$\beta = 309/19 \text{ s}^{-1} = 5 \text{ s}^{-1}$$

$$T = 20$$

$$\bar{T} = \frac{1}{\beta - 1} = \frac{6}{20} = 0.3$$

$$\bar{T} = \frac{1}{B' d} \Rightarrow B' = \frac{1}{\bar{T} d} = \frac{13}{6} s^{-1}$$

$$B \left\{ \begin{array}{c} X\% \\ \hline B' \\ \hline B \end{array} \right. \quad x = 1 - \frac{B'}{B} = 0,56 = 56\%$$

ZAD. H₂O - SINKRONIZACIJA OCETRO VODIMA I VISINA PECI STVORNOG MOLERIKA VOZI

dretva atom (X)

{

mutex-lock(&M); // X = ili 0 ili H

dok je ((X == H) && br[H] == 2)

// ((X == 0) && br[0] == 1))

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

br[X]++;

ako je ((br[H] + br[0]) < 3)

cond-wait(&shllopaju, &M);

inace { // == 3

shllop - molokolo();

} cond-broadcast(&shllopaju);

br[X]--;

cond-signal(&red[X]);

}

ZAD. 7.5. U NEJOM POSLUŽIVELJSNOM CENTRU NAPRAVLJENA JE ANALIZA RATA POSLOVITVONA. USTANOVLJENO JE DA TAI POSLUŽITELJU RATE S POBRALJENIM MOLIM OPTIJENJOM, POSLUŽITELJU P1 PROSJEČNO DOBIVA 70 ZADUŠAKA U MINUTI I NJEGAVA PROSJEČNA ISPRAVOST JE 20%, POSLUŽITELJU P2 DOBIVA 200 ZADUŠAKA U MINUTI I NJEGAVA PROSJEČNA ISPRAVOST JE 30%, DOK POSLUŽITELJU P3 S PROSJEČNO 150 ZADUŠAKA U MINUTI RADI S 70% OPTIJENJOM. POSLUŽITELJU P3 JOU PROSESOPASNOST JESTI 50% JESTI JE P1 70 70 100% JESTI JE P2. UZETIJUJIH KVALITATU USLUGE (PROSJEČNO VRIJEME ZADUŠIVANJA ZADUŠAKA U SUSTAVU)

Ukoliko bi se svu poslovnu poslovcu na poslužitelju p3.

$$\begin{aligned} P_1: L_1 &= 70 \text{ min}^{-1} = \frac{7}{6} \text{ s}^{-1} \\ g_1 &= 20\% = 0,2 \end{aligned} \quad \left\{ \begin{array}{l} \beta_1 = \frac{7}{6 \cdot 0,2} \end{array} \right.$$

$$\begin{aligned} P_2: L_2 &= 200 \text{ min}^{-1} = \frac{20}{6} \text{ s}^{-1} \\ g_2 &= 30\% = 0,3 \end{aligned} \quad \left\{ \begin{array}{l} \beta_2 = \frac{20}{6 \cdot 0,3} \end{array} \right.$$

$$\begin{aligned} P_3: L_3 &= 150 \text{ min}^{-1} = \frac{15}{6} \text{ s}^{-1} \\ g_3 &= 10\% = 0,1 \end{aligned} \quad \left\{ \begin{array}{l} \beta_3 = \frac{15}{6 \cdot 0,1} \end{array} \right.$$

$$\beta_{3,1} = 1,5 \beta_1$$

$$\beta_{2,1} = 2 \beta_2$$

$$g'_3 = g_3 + \frac{\beta_1}{1,5} + \frac{\beta_2}{2} \approx 38,3\% \quad \text{ali} \quad g'_3 = \frac{d_3}{\beta_3} + \frac{d_2}{\beta_{2,1}} + \frac{d_1}{\beta_{3,1}}$$

$$d = d_1 + d_2 + d_3$$

$$\beta = \frac{d}{g'_3}$$

$$\bar{T} = \frac{1}{\beta - d} = 0,089 \text{ s}$$

GOSPODARENJE S POMEMNIČKIM PROSTOROM

* PROCESOR: GENERIŠE ADRESĘ NA 3 NAČINA:

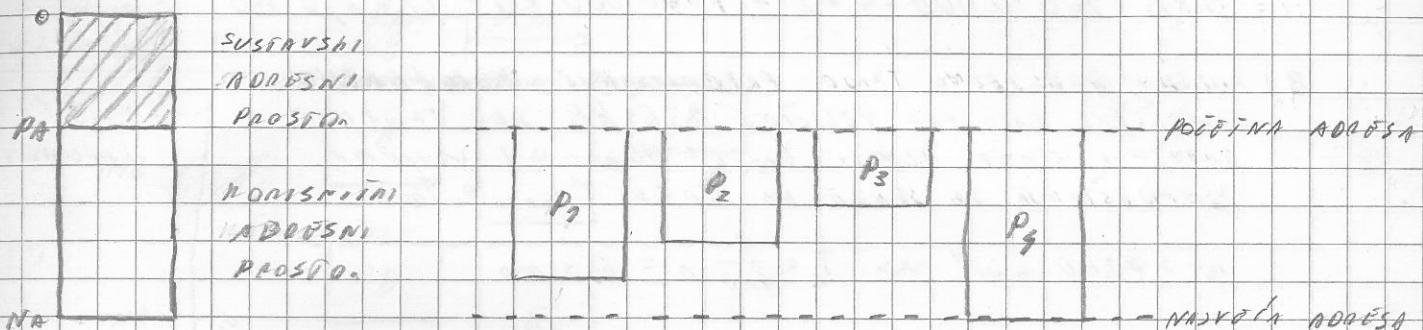
1. ADRESĘ INSTRUKCIJA MOže DOŁAĆ 12 PROGRAMSKIH BROJALI

2. ADRESĘ ODREDIHO I POKLADAT OPĆINU

3. ADRESA POKLADNE SLOBE (BROJ SVEZAKA BKO POKLADNIKA)

* FIZIČKI PROSTOR JE MANJI IZMJEĐU LOGIČKOG PROSTORA

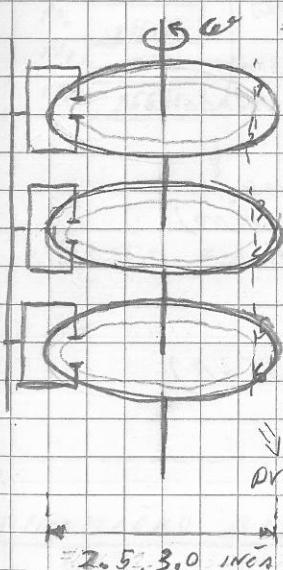
* ZAHTIJENI ADRESNI PROSTOR SADEŽI "PAZAN 'D10'" DA MOŽE RAST.



OSNOVNA SVRŠJAVA MAGNETSKIH DISKOV

* MOŽE SPOREMITI VOLJENO POLJEĆE MEMORIJE

- SPONI, OLI NAVJEĆUJU S OBZIROM NA CENU NO JEĐINICI MEMORIJE



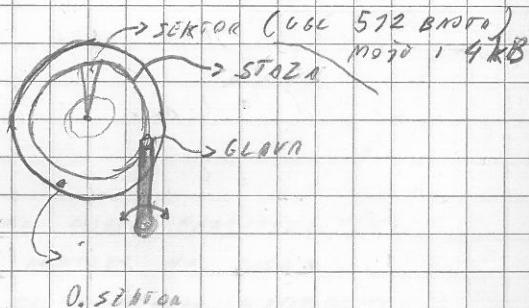
DIREKTNE

= 2.5, 3.0 INČA

! ŠARF STOZA JE PONOKE
RADIJUS = CILINDAR

! SEKTOR SE ADRESIRA:

1. DEONI BROJ PLOŠIB (1 SLOV)
2. DEONI BROJ SEKTOA NA PLOŠI
3. DEONI BROJ SEKTORA NA SEKTOU



* BRZINA ČITANJE I PISANJE

JE JEĐINICA (U PARALEL

JE ČITANJE MOTO (RPM)

=> JEĐINICA BRZO SE VRI

UKUPNO TRAJANJE PONENOSA PODATAKA

* TRAJANJE POSTAVLJANJA GLAVE (HEAD POSITION TIME)

→ TRAJANJE TRAJANJE STOZE (SEEK TIME)

1. UBOZAVANJE
2. GIBANJE KONSTANTNOM BRZINOM
3. USPODARAVANJE (KEDONJE)
4. PLENO PODEŠAVANJE

→ ROTACIJSKO PONJENJE (ROTATION latency) = $T_{rot}/2$

* TRAJANJE PONENOSA PODATAKA (DATA TRANSFER TIME)

→ TRAJANJE ČITANJA DISKOLO IZ CIČIĆA STOZE

→ TRAJANJE PROMIŠTANJA GLAVE SA STOZE NA STOZU

* POSTOJE DO NEKADA I PANTOP POKRETANJE (DISK SE VRASI 000 TI NESTO ČAS)

Zadatak 8.2. Disk ima 700 sektora po stazi, a skupno je ukljuceno 1kB, 4 ploče i novi so bezimao $\omega = 7200 \text{ okretaja/min}$. Podataci su zapisani na obe strane ploče. Uporavljaci skup pročita jednu cijelu stazu u intervalu spremnika, a zatim prenosi posebne sektore ili cijelu stazu u glavni spremnik. Prijenos u glavni spremnik dovira sa brzinom 20 Mbit/s , a za to vrijeme sklop može čitati 5 diskova. Disk ima 2000 staze.

* fad prijenos
o brzini
 $M = 10^6$
(debljina)

A) Koliki je kapacitet diska?

$$1 \text{ kB} \cdot 100 \cdot 2000 \cdot 4 \cdot 2 = 1600000 \text{ kB} = 1562,5 \text{ MB}$$

* prijenos vrijem
diska je
trenutno vrijeme

B) Koliko potreban je razmak između pozicija
smještenih datotičkih vrijednosti 135 kB da
razmak između pozicija 12 nosi $t_{seek} = 10 \text{ ms}$ i vrijeme
premještanja sa stazom na stazu $t_{travel} = 1 \text{ ms}$

→ Nije

Prijevozni razm

$$\omega = 7200 \text{ min}^{-1} \Rightarrow T_r = \frac{1}{720} \text{ s} = 8,3 \text{ ms}$$

Prijenos datoteka: 1. pronalaženje datoteke

$$t_{seek} + T_r/2$$

2. čitanje cijele staze - T_r

3. 3) prijenos podataka (prijenos)

$$t = \frac{1}{2} (S - Brzina binarni staze)$$

$$t = \frac{100 \cdot 1025 \cdot 8}{20 \cdot 10^6} \text{ s} = 40,96 \text{ ms}$$

b) premještanje glave na prvu stazu

\Rightarrow u ovom slučaju ovo nije potrebno

jer imamo 4 ploče

$$\Rightarrow \text{max} \{ t_{seek}, t_{travel} \}$$

(jer je zbiraju paralelno)

4. rotacijsko usporjavanje

$$T_r/2$$

5. čitanje cijele staze - T_r

$$T_r \text{ (ne može proći pre 30 ms = 35,68 ms)}$$

6. prijenos podataka

$$35\% \cdot t_{prijenosa} \text{ (samo } 35 \text{ kB)}$$

$$t = \frac{35 \cdot 1025 \cdot 8}{20 \cdot 10^6} = 14,37 \text{ ms}$$

$$\sum = 90,3 \text{ ms}$$

* pogledaj u narednim zadatcima 8.9.

! Tho u posljedu modu generirat adresni program!

1. programski brojilo (PC)

2. instrukcija

3. rezultatni skloplj (RA)

PROCESNI INFORMACIJSKI BLOK

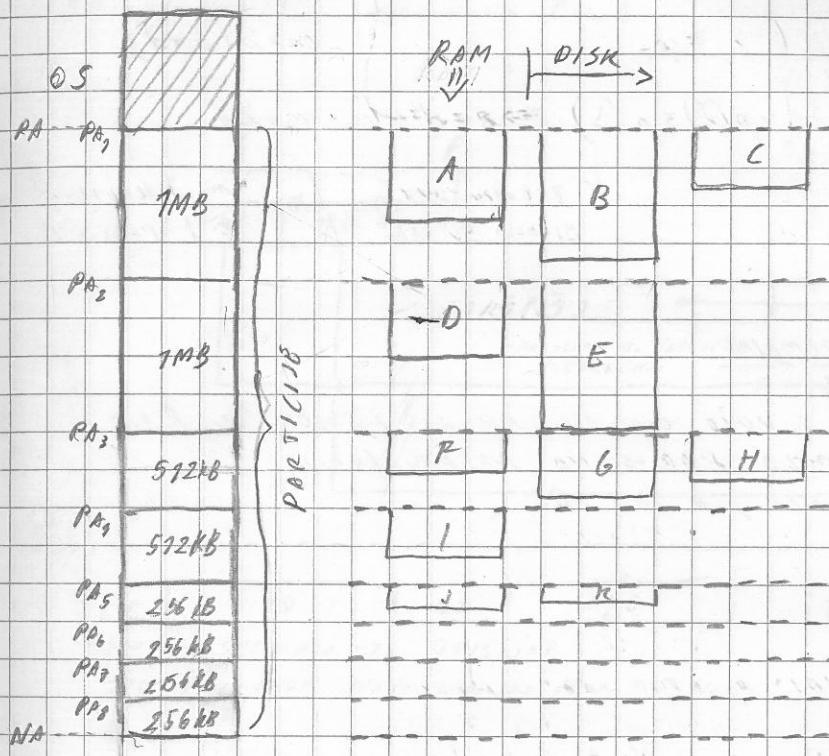
* "OPISNIK PROCESA"

- SHAP PODACIMA KOJI OPISUJU PROCES

- SNOŽNI PODATCI:

- A) GOIB SE PODACI NALAZI U RAM-U, NA DISKU I VIRTUALNOJ MEMORIJI
- B) OSOZI PODACI O PROCESU (PID, GOIB SO DATOTEKE I GOST...)

SUSTAVSKO RASPREDJIVANJE SPOMINJA



- VRSTIĆE FRAGMENTACIJE: - UNUTARNA - PROGRAM ZAVRIMA CHOLU PRAVICIJI
 - SISTEM NEĆE SE KOMITI NA DISKU
 - VANJSKA - KADA JE NEKA OD PRAVICIJA SLOBODNA, A PROGRAM NIJE SV POKRENUTI ZA DRUGO PRAVICIĆE ČEKAZU NA IZVJEŠTAJ

DINAMIČKO RASPREDJIVANJE SPOMINJA

- DODJELIĆE MEMORIJE KODNO PROCESU TIGBO

=> KAO PROCES ZAVRŠI NESTANO KUPN => PRAGMONTACIJA

• PREGO MOŽIMO SPOJITI

• NOVOM PROCESU MOŽETI DODIJELITI NEJMENI POKLJUBNU KUPN

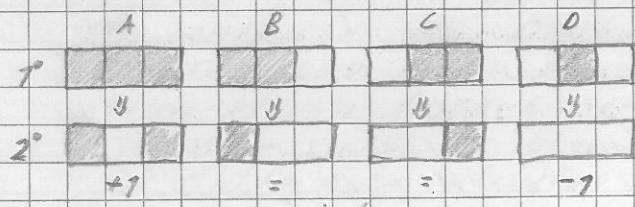
! FRAGMENTACIJA SÖ NE MOŽE IZBEGAVIĆI, VOLI SAMO UBLJADJIVI

- KOMPILERI SE: - za dodjelu adresnih prasaga (VLSI MATERIJA)

- za upravljanje na razini OS-a

- za upravljanje 60 milij (HASP-a) UNIVID PROCESA

• KNUTHORO 50% PRAVILO



(BROJ RUPA U SEKCIJARNOM STANJU JE 20% 50% BROJA POPUNJENIH DOLINA)

(PRAVILO)

$$m = a + b + c + d \quad (\text{BROJ RUPA BLOKOVU})$$

(PRAVILO)

(PRIMJER U BROJU RUPA)

ISLJEČAK

$$b = c$$

$$z_{ij} = \frac{2d+b+c}{2} = b+d$$

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

$$p(P) = p_0 \frac{a}{m}$$

$$p(\lambda) = p_0 \frac{d}{m} + p_2 q$$

$$\left. \begin{array}{l} g \ll 1, p \approx 1 \\ p(P) = p(\lambda) \Rightarrow a = d \end{array} \right\}$$

$$\left. \begin{array}{l} p_0 = p_2 \\ m = 2(b+d) \end{array} \right\}$$

$$m = 2h$$

* p_0 - VJEĆNOSTI OSLOBOĐENIA

p_0 - VJEĆNOST ZAVRŠINJA

TROJINA SVIH
BLOKOV SU RUPA

$$\boxed{n = \frac{m}{2}} \quad \begin{array}{l} \text{KNUTHORO} \\ \text{PRAVILO} \end{array}$$

PREKLONI NOĆIM UPOGOBE RAM-OG SPREMNIKA

- PROGRAMER SAM PODIJELI MEMOIJU I VUĆE ONO ŠTO MU JE POTREBNO U RAM
- RAM SE KAO SUSTAV DI IMAMO MALO RAM-A NA RASPODABLJAVU
- TA ČTO SE USKOVARI BRAĆE PODIJELI, A NE OS

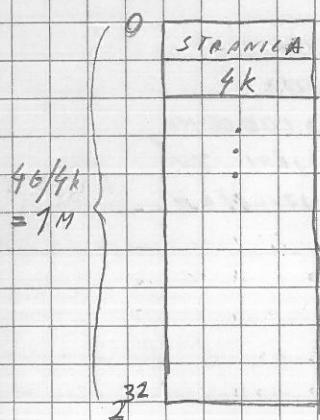
STRONIĆENJE (PAGING)

- PODIJELI LOGIČKI, FIZIČKI I VIRTUALNI PROSTOR NA KLASTORE

Logički zdr.
prostor

Fizički zdr.
prostor (RAM)

Virtuelni zdr.
prostor (HDD/SSD)



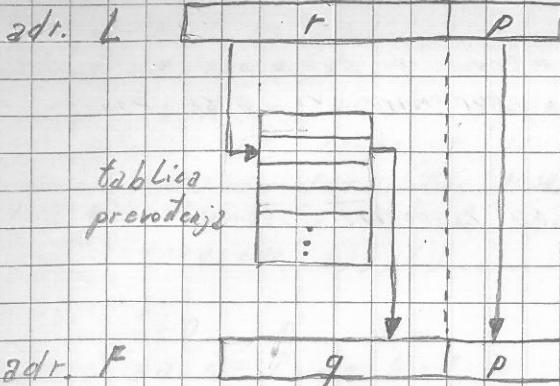
* KLASTOR
= NAKLJUČNA SLOGA

P - BROJ BLOK UNIKTOR
STRANICOI
 $= 12$ (32bit & 64bit)

P - BROJ BLOK UNIKTOR
OKVIRA
 $= 12$

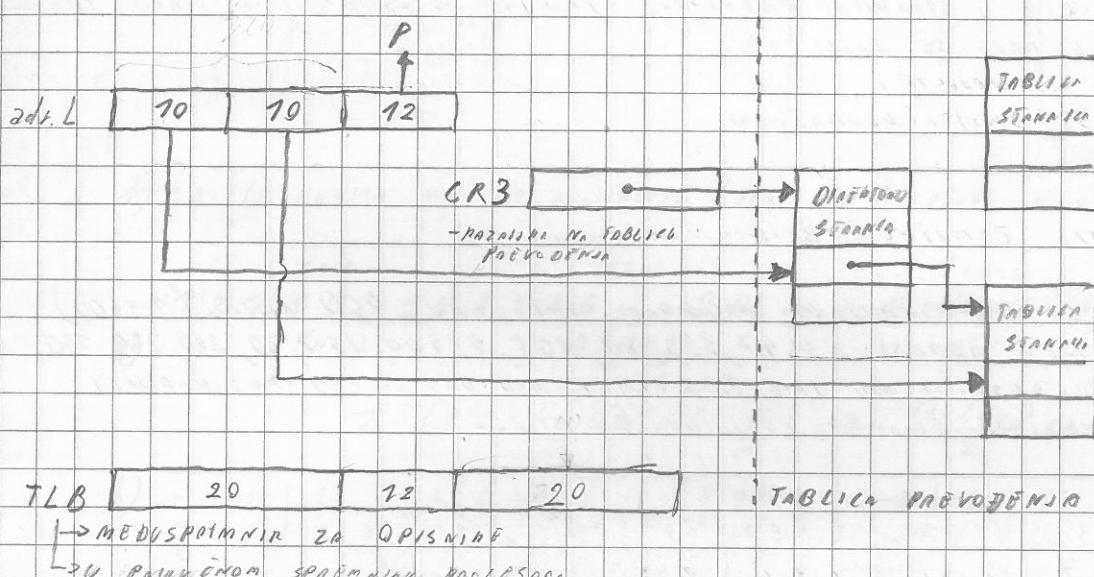
t - BROJ BLOK STRANICE
(OSLOBOĐENI BITI)
 $= 20$ (32bit)
 $= 52$ (64bit)

q - BROJ LIO OKVIRA
POKRIĆA 1MB RAM-NE
RAM RASPODABLJAVI
 $- q \leq t$



! TABLICA PRENOVNIJA POREVODI LOGICHU U PIZICNU ADRESU
 - IDEJMO SAOPZI q I BIT PRISTUPNOSTI
 - U PONASI JE TO VISE TABLICA

* TABLICA PRENOVNIJA SE SASTOJI OD
 DEONOG DIREKCIJSKOG STRANICA /
 BLOKOM JEDNE TABLICE STRANICA
 (DIREKCIJSKI STRANICA IME "9"
 nAZIVNI, A TABLICA STRANICA
 1 K ADRESA)



• STRANČENJE NO ZAHTEVU (DEMAND PAGING)

! PREKID ZBOG PROMAJSAJU STRANICU SE RAZCINJUJE OD DRUGIH PROSESIMA:
 - PREKID SE NOVODA VRBED IZVOĐENJA INSTRUKCIJE
 - INSTRUKCIJA ŽE MOže PONOVITI
 - ZNAJI SAKOZNAJ SVIH NÖGISTORU PONJE IZVOĐENJA

- ① ADDRESIRANJE NEPRISTUPNE STRANICE
- ② PREKID ZBOG PROMAJSA
- ③ PONOVITI STRANICU NA DISKU
- ④ PODEŠAVANJE STRANICE U PRAZAN DAVIR
- ⑤ AŽURNANJE TABLICE STRANICA
- ⑥ PONOVNO POKRETANJE DRITVU

* U PONASI POCESOM
 IMAJU SPALJENE
 REGISTRE KOJI SU
 "SJENA REGISTARA"
 => B NJIMA SE NE
 MOže DESITI PREKID
 I AKO SĘ SVE
 ZBESI... BH

• STRATEGIJE ZAMJENE STRANICA

- ČISTO STRANICA - STRANICA POJI SU TIDOM IZVODENJA PROGRAMA NEĆE BITI UVEĆANA
- NEČISTO STRANICA - MIJENJALA SE PA SE TREBBI SPREMIO NA OŠTRU POGUZU

* U ARHITEKTONI Intel x86 DODAJI 12 BLOKA SV ZASTOVICE STRANIC

REDNI BROJ	OKVIR	GL	D	A	W _f	W ₀	W _V
31	12 11 10 9 8 7 6 5 4 3 2 1 0						

V - BIT PAZURNOSTI U REDOVOM SPREMNIKU

W - POSTAVLJENA ZASTOVICA ZABRANjuje PISANje U STRANICU

O - STRANICU VPODREDOVJAJA OPERATORSKI SUSTAV

W_f - SUPRO PISUNje U STRANICU PAZISUJE STRANICU IZ PRIPUĆNEG U GLAVNI SREDINIK

A - STRANICA U KOJU SÖ PAZISTVUJE

D - STRANICA JE NEPIST

GL - STRANICA SÖ SMETA GLOBALNOM

• TEORIJSKO STRATEGIJE ZAMJENE (IZBACIVANJE) STRANICA

ZAD. 8.3 U SUSTAVU SA STRANICOM program VOLJENO 400 redki (1-400) GORE NAKON SLIČNI ADRESI: 23, 57, 333, 81, 105, 7, 400, 157, 30, 209, 242, 360. 10. program na raspodjeljenu 1mn 200 redki pravog straznica. MNOZATI NIZ PET PREDVIĐENJA STRANICA VOLJENO 50 redki

adresi	23	57	333	81	105	7	400	157	30	209	199	360
stranica	1	1	7	2	3	7	8	4	7	5	3	8

FIFO	1	1	1	1	1	1	1	1	1	1	1	1
	7	7	7	7	7	7	7	7	7	7	7	7
	2	2	2	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3	3	3	3
LRU	1	1	1	1	1	1	1	1	1	1	1	1
	7	7	7	7	7	7	7	7	7	7	7	7
	2	2	2	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3	3	3	3
OPT	1	1	1	1	1	1	1	1	1	1	1	1
	7	7	7	7	7	7	7	7	7	7	7	7
	2	2	2	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3	3	3	3
LFU	1	1	1	1	1	1	1	1	1	1	1	1
	7	7	7	7	7	7	7	7	7	7	7	7
	2	2	2	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3	3	3	3

više mogućih
rijanja

više mogućih
opravljanja

"Optimalna strategija (OPT) ne može biti implementirana,
ali se koristi u praktici"

ZAD. 8.3 U SUSTAVU S VJEĆA U NIM SPECIJALNIM SPREMAMA VJEĆEĆNA DAKIRA JO N PREDI, A ODRIM
JO PUNE NO ZAHVATI. ALGORITAM ZAMJENE STAVNIKA JE LRU.
POZIČAK A [1..N, 1..N] JE PO PODGLIMA (SA SUSPENZIJOM LOKACIJOM
38 MIJESINA DESNE INDEJSKE).
KOLIKO ČE PREDVIĐENI IZAZIVATI PREDMETNO PROGRAM ZA PREDSTAVLJANJE A-
U RAVNOM SPREMNIKU POSTOJI: 1 ODRIM; 2 ODRIM; 3 ODRIM; NODIM?
(ZAHVATI PREDVIĐEĆE CIGLE DOMINATA NAIROBI I PREDSTAVA PREDMETNIM
KONZABLOM CIJETI)

$$\left. \begin{array}{l}
 t = 0 \\
 z_2 \quad i = 1 \text{ do } N-1 \\
 z_2 \quad j = i+1 \text{ do } N \\
 \{ \\
 t^+ = A[i, j] \\
 t^* = A[j, i]
 \end{array} \right\} \quad \left. \begin{array}{l}
 A = \begin{bmatrix} 1 & 2 & \dots & N-1 & N \end{bmatrix} \\
 \{ \\
 1 \\
 2 \\
 \vdots \\
 N-1 \\
 N
 \}
 \end{array} \right\}$$

$\Rightarrow N_{\text{Oxide}} \Rightarrow N_{\text{Promesa}}$ (por lo $N_{\text{Promesa}} \neq N_{\text{Oxido}}$)

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

$$b) \quad \left| \begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c} 1 & 1 & | & 1 & \dots & | & 1 & 2 & 2 & | & 2 & \dots & | & 2 \\ -2 & 3 & | & 3 & & | & N & N & 3 & | & 4 & \dots & | & N \\ \hline N & & | & & & | & N-1 & & & | & 4 & & | & 3 & | & 1 \end{array} \right.$$

$$N + N - 1 + \dots + 9 + 3 + 1 = \sum_{i=0}^N i - 2 = \frac{(1+N)}{2} \cdot \frac{N}{2} - 2$$

$$\begin{array}{c}
 \text{c)} \quad -1 & 1 & 1 & 1 & 1 & 3 & 3 & \dots & N-1 & -3 & -3 & -3 & -3 & -1 \\
 -2 & 2 & 2 & \dots & N-2 & 2 & 2 & \dots & 2 & N+ & -2 & -2 & N & N & N \\
 -3 & 3 & N & N & N & 4 & N & \dots & N & N & N & -1 & -1 & -2 & -2 \\
 \hline
 * & & & & & & & & & & & & & & \\
 N & & & & & N-1 & & & & & & & 4 & 2 & 0
 \end{array}$$

$$N + N - 1 + \dots + 4 + 2 = 0 = \sum_{i=1}^N i - 3 - 1 = \frac{(1+N)}{2} \cdot N - 4$$

* OSTVERIVENJE LRU JE NEFRIMASNO JER BI TABBALI KORISNIKI TONU DOSENIO
 MUDOMA IZBACI BI = MOT POSMINO REGISTRI U SVAKU STANICU
 \Rightarrow U INFOL X86 AČETIKOVATI BI MOGUĆ "ZD TO KORISNIKI
 D I A BLOIRE (LASE JE A BIFNJI) ZA BONAKIE
 OSVRAOTI REGISTRI
 \Rightarrow U PAKASI JE KORISNI SATNI MITIGANIZAM

• SATNI ALGORITAM

- INICIJA LRU ALGORITMA
- PRAVILJENO ZA JEOPROGOLODOŠI SUSTAV JER SVAKI PROSESIMA IMA VELASNOVU CESTU
(ZA VISOPRVOLODOŠI SUSTAV SE U PRVOM POKUSU RIŠE)
- NEPODARI SE KUVĀNO LISTO I. 100 SE PO NJUJ GLEONOVU BIG A
- $1 \Rightarrow$ PREBRIŠI U 0
- $0 \Rightarrow$ MOŽEŠ PISAT PO FORMI

• GOSPODARENJE OKVIRIMA

- STONJA OKVIRI: - AKTIVNI - KORISTI SE
- SLOBODNI - NO KORISTI SE, ALI SU TREBA DOBACIĆI SREDJU PRED PROJEKT
- SLOBODNI S OBRAZINIM SADRŽAJEM - SPODNEVNI ZA DODJELU
- NEISPRAVNI (PO PODNEBI POGODI) - GORE O SREDJU

ZAD. 8.5. U NEKOM SUSTAVU TREBAJU SE OBAVIOVI 4 PROGRAMI KOJI SU VED
PRIPAPLJENI NA POMOĆNOM SPOMINU I ZAUZIMAJU 5 MB, 8 MB, 3 MB, 10 MB.
DODODNO JE PODZET 1 ZAUZETAK PROGRAMA ZVANIĆI UNAPRIJEĐ
MOGU SE ISPOSTAVITI SLEDEĆIM NIZOM DOGOĐAJA:

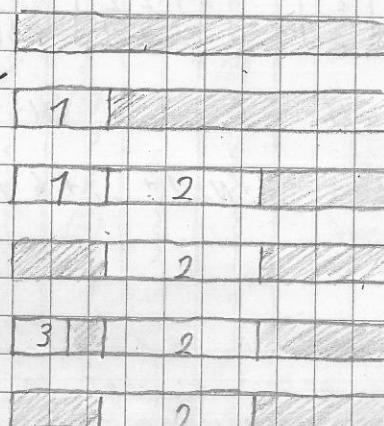
ZAUZETAK	OSLOBOĐANJE
t_1	P_1 (5 MB)
t_2	P_2 (8 MB)
t_3	P_1 (5 MB)
t_4	P_3 (3 MB)
t_5	P_3 (3 MB)
t_6	P_4 (10 MB)
t_7	P_2 (8 MB)
t_8	P_4 (10 MB)

SUSTAV MU RASPOLOGOJI 100 20 MB
MEMORIJU REZERVIRANO ZA PONOSNIKE PROGRAMA
PRAVILJENO STONJI KONAK SREDJU AKO SE
PODSTRE MULDE UPRAVLJANJA SPREMNAKOM:
a) STONIKA, SEGMENT 00 70 MB
b) DINAMIKO
c) SREDJU, STEANICA 00 1 MB

a)



b)



P_4 ne stane!

c)

 t_0 t_1 1 1 1 1 1 t_2 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 t_3 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 t_4 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 t_5 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 t_6 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3| 2 2 2 2 2 2 2 2 | 4 4 4 4 | \Rightarrow obriši nepotrebne

| 4 4 4 4 4 2 2 2 2 2 2 2 2 4 | 4 4 4 4 |

 t_7 1 4 4 4 4 4 2 2 2 2 2 2 2 4 | 4 4 4 4 | t_8 1 4 4 4 4 4 2 2 2 2 2 2 2 4 | 4 4 4 4 |

ZAD 8.6. SATNI ALGORITAM

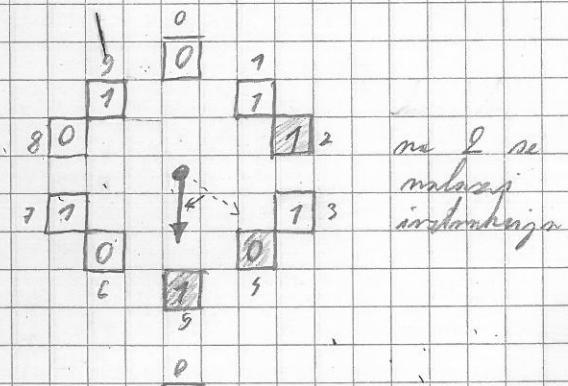
ZA NEKI SUSTAV NOJI KONFIG STANJE IZMJEŠE, ZADANA JE TABLICA PREDVODENJE ZA JEDAN PROSES, STONJE OPVJERA SUSTAVA, STANJE PRAZNIK I ZASEVNICA A (za upravljanje međusobnim sotnjem algoritma). Ako zadani procesos treba napraviti zadano instrukcije kada će se sotnju mijenjati? Neka se istaknute naloze u o. stanici

TABLICA PREDVODENJE:

STANJE	OPVJA	(P)bit prikazanost
0. 99	2	1
1. 100	0	1
2. 200	0	0
3. 329	6	1
4. 329	0	0
5. 600	0	0
6. 599	0	0

INSTRAUKCIJE:

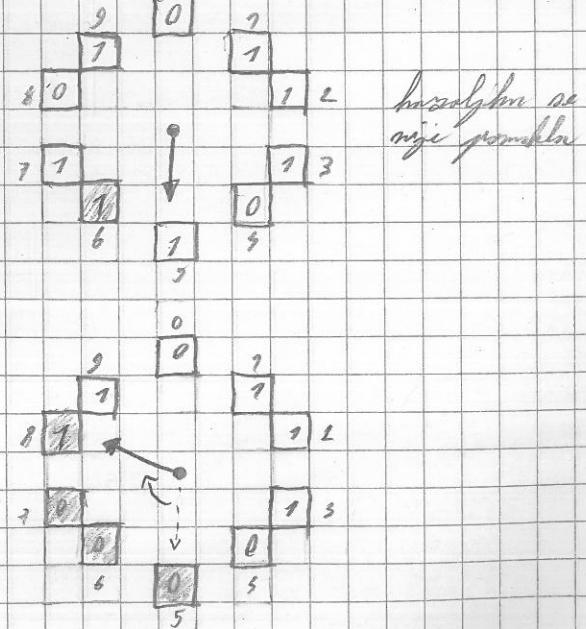
1. LDR R1, (508)
2. LDR R2, (332)
3. LDR R3, (256)



STANJE PRAZNIK I ZASEVNICA:

OKVIRI:

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

X - stanica
neke drugega
mreža