**1. ZADATAK:**  
Ako pristignu 3 signala iste vrste za redom (ne istovremeno), a obrada traje jako dugo, onda će:  
A) Odraditi sve signale po redu kako su dolazili  
B) Odraditi sve signale obrnutim redosljedom  
C) Odraditi samo 2 signala  
D) Odraditi samo prvi signal  
  
**ODGOVOR: C**   
kada dođe prvi, krenut će ga odrađivat, onda će stići drugi i zapamtit će da treba i njega odradit, onda će stići treći, a ovaj još odrađuje prvi, a već ionako čeka jedan, pa ovaj treći neće registrirat. nakon što se odradi prvi, odradit će se i drugi jer je njega zapamtio.  
  
**2. ZADATAK** (\*nisam siguran jesam li točno prepisao kod)

for (i=0,t=0; i<5; i++) {

if (fork () == 0) {

proces (&t);

exit(0);

}

}

Koje tvrdnje nisu točne?  
A) Stvorit će se 5 novih procesa  
B) Stvorit će se 5 novih dretvi, svaka u svom procesu  
C) Procesi mogu komunicirati pomoću "t"  
D) dretve mogu komunicirati pomoću "t", ali procesi ne mogu  
  
**ODGOVOR: C i D tvrdnje nisu točne**   
prva se kao podrazumijeva, druga je -kaže Jelenko- trik-pitanje, jer se dretve naravno naprave automatski čim se stvori novi proces, treća nije točna jer svaki proces ima svoj "t", a d je -kaže Jelenko- nebuloza.

**3. ZADATAK**

int i=0, a=0;

void \*dretva (void \*x) {

if (i&1) i++;

else a++;

return NULL;

}

int main () {

for (i=0; i<5; i++) {

pthread\_create (&ids[i], NULL, dretva, &i);

sleep (1);

}

printf("a=%d\n",a);

return 0;

}

Kolika je konačna vrijednost varijable "a"?  
Koliko će dretvi stvoriti glavna?

**ODGOVOR: a=1; 3 dretve**   
na i=0 stvori dretvu br I. "&" operator je logički i istinit je ako je broj neparan. U prolazu i=0, izraz (i&1) nije istinit, pa je a++, tj. a=1. Nakon toga izađe, vrati se na početak for petlje i poveća i. Sada je i=1. Stvori dretvu broj II. Izraz (i&1) je istinit, pa i++, tj. i=2. Vrati se u for petlju, pa poveća i na 3. Stvara dretvu broj III. Izraz (i&1) je i dalje istinit jer je i povećan za 2 od zadnji put, pa će još jednom i++, tj. i=4. Vrati se u for petlju, pa ga poveća na 5. i=5. (i<5) više ne vrijedi, pa izlazi iz for petlje i -kraj programa-. a je ostao na 1, a 3 dretve su stvorene.

**4.**  Imamo globalnu varijablu A, koja se u glavnoj (main) funkciji postavi vrijednost nula. Nakon toga kreiramo 50 dretvi koje će raditi sljedeće:

html kod:

for(i=0; i<10000; i++) A=A+1;

Koja će biti vrijednost varijable A nakon što sve dretve obave svoj posao?  
  
a)A=500000  
**b)A<=500000**  
c)A=>500000  
d)A<500000  
e)A>500000  
Objašnjenje: *"Naš je prof spominjao da bi dretve mogle (kojiput) uzeti istu vrijednost varijable (dakle pristupe podatku u isto vrijeme... prije nego je ona druga spremila uvećanu vrijednost) i uvećati za 1 (efektivno, samo je jedna dretva uvećala varijablu, a ne više njih). Dakle, pod b) bi bio točan odovor."*  
  
**5.** Procesu se pošalju tri signala sljedećim redoslijedom: SIGQUIT(1), SIGINT(2), SIGINT(3), od čega se svaki obrađuje mnogo dulje nego što mu stigne sljedeći.  
Kojim redoslijedom će se prekidi obrađivati?  
a)  
b) SIGQUIT(1), SIGINT(2), SIGINT(3) (redoslijedom kojim su i dovedeni)  
c)   
d) SIGINT(3), SIGINT(2), SIGQUIT(1) (redoslijed suprotan od onog kako smo ih doveli)  
e) drugo:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
**TOČNO:** SIGQUIT(1) se počne obrađivati, SIGINT(2) ga prekine i počne se obrađivati, SIGINT(3) dođe ali vidi da se već obrađuje SIGINT(2) koji je isti kao on  
Nakon što SIGINT(2) završi, kreće SIGINT(3), i kad on završi, SIGQUIT(1) nastavi do kraja.  
  
  
**6.** Korištenje proces u odnosu na dretve:  
a) preporuča se jer ... ne znam kak je bilo dalje  
**b) ne preporuča se jer procesi zauzimaju više memorije sustava**  
c)  
d)  
  
  
**7.** Nakon stvaranja procesa djeteta, koje su promjene varijabli koje učini proces roditelj vidljive djetetu  
a) Varijable roditelja koje se prenesu u dijete  
b) Globalne varijable  
**c) Samo one koje se nalaze u zajedničkoj memoriji**  
  
**8.** Višedretveni rad procesor obrađuje tako da  
a) dretve se obrađuju slijedno - kako jedna završi, druga počinje  
b) sustav određuje koja će se petlja sljedeća izvršavati  
c) odrađuju se odjednom  
**d) za svaku dretvu stvara se virtualni procesor i taj virtualni procesor obrađuje tu dretvu...**  
Objašnjenje (iz uputa za labose): " *LWP jest osnovna dretva upravljanja na razini jezgre sustava. LWP se, sa strane programera, može predočiti jednostavno kao* ***virtualni procesor****.*"

**9.** Imamo globalnu varijablu A=0, glavni proces napravi 100 procesa. Svaki proces u svom radu 100 puta poveca varijablu A za 1. koliki ce A ispisati glavni proces?  
- 100  
- 1000  
- 10000  
**- 0**  
- >100  
  
**10.** Sto vraca fork nakon uspjesno izvedenog forkanja ?  
-puno pogresnih odgovora i naravno tocan: **roditelju PID djeteta, a djetetu 0**   
  
**11.** programski odsjecak koji stvara dretve, i samo se tu stvaraju dretve:  
for (i=0; i<5; i++)  
pthread\_create (&tid, NULL, dretva, &i);  
pthread\_join (tid, NULL);  
  
-svaka dretva dobije svoju vrijednost i  
**-na kraju ce se cekati samo zadnja stvorena dretva**  
-nece se uspjeti napraviti dretve jer nije dobro zadano  
-Istovremeno ce max 2 dretve biti u procesu

**12.** Na sto treba paziti kod visedretvenih programa?  
- na lokalne varijable  
**- na globalne varijable**  
- stog  
- nista jer D\_REENTRANT pazi na to  
  
**13.** u programu je signal SIGQUIT maskiran, a SIGINT nije. ako za vrijeme izvodjenja SIGQUIT-a dodje signal SIGINT sto ce se dogoditi?  
- SIGINT ce se zadrzati  
- ???  
- ????  
**- SIGINT prekida cijeli program**

**14**. U nekom sustavu maskiran je SIGQUIT , javi SIGQUIT 3x zaredom u razmaku od 1 sekunde , a njihova obrada traje znatno dulje od 1 sekunde , što će se obraditi tj kako će se obraditi  
**Obrade se samo prva dva signala , s tim da prvo prvi pa zatim drugi (3. se ignorira)**

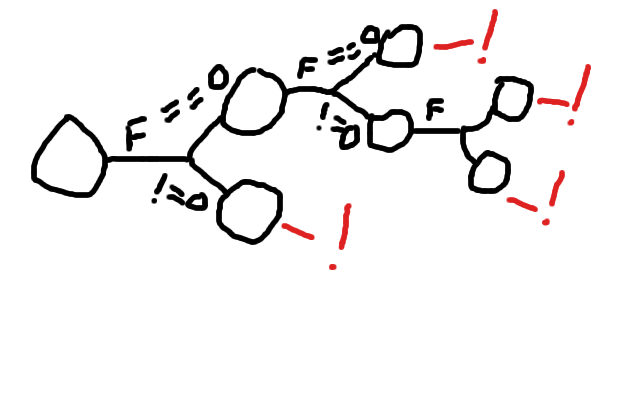
**15.**  Zadan je dio  
For (I=0, t=0, i<5; i++)  
If (fork())==0)  
Proces(&t);  
Exit(0)  
  
**- Stvara 5 novih procesa  
- Svaki proces ima u sebi barem jednu dretvu  
- Procesi NE MOGU komunicirati preko varijable t  
- Niti dretve NE MOGU komunicirati preko varijable t**  
  
**16.** Int i=0, a=0  
void \*dretva(void \*x)  
Int n= \*((int\*)x)  
If (i&1) i++  
Else a++  
Return NULL  
Int main()  
For (i=0; i<5; i++)  
Pthread\_create(&ids[i], NULL, dretva, &i)  
Sleep(5)  
Sleep(10)  
Printf(„A= %d\n“,a)  
Return 0  
  
Trace :  
i a  
0 0  
0 1  
1   
2 1  
3  
4 1  
5  
„end“  
  
**A=1, broj dretvi = 3**

**17.**

if(fork()==0)  
if(fork ()!=0)  
fork ();  
 printf("!");

ispise se '!' 4 puta

OBJAŠNJENJE



**18.**

fork();  
fork();  
pthread\_create(.).)  
pthread create (...)

ODGOVOR:

na početku imaš 1P + 1D, nakon 2 forka 4P + 4D i onda svaki proces stvori još 2 dretve (uz onu jednu početnu) = 3D/P \* 4P = 12D

**19.**

fork();  
printf("a");  
fork();  
printf("b");  
fork();  
printf("c");  
koliko puta ce se ispisati slova a, b ,c -🡪**tocan odgovor**=>2a, 4b, 8c

**20.**

**Int i=0**, a=0  
void \*dretva(void \*x)  
Int n= \*((int\*)x)  
If (i&1) i++  
Else a++  
Return NULL  
Int main()  
For (i=0; i<5; i++)  
Pthread\_create(&ids[i], NULL, dretva, &i)  
Sleep(5)  
Sleep(10)  
Printf(„A= %d\n“,a)  
Return 0

Koliko puta će se A ispisati?

A=3 i pet dretvi bi se stvorilo kad "i" ne bi bio globalna varijabla, nego varijabla unutar maina

**19.**

A=0;  
fork();  
A++;  
if(fork())  
A++;  
printf("A=%d\n", A);  
  
  
isprinta se : 2 1 2 1

**20.**

while(fork()==0);  
printf("obrada zavrsila");  
  
sto bi se dogodilo/ ispisalo i koliko puta?

beskonačno novih procesa jer imaš bestonačnu petlju, i svaki bi proces ispiso obrada završila

**21.**

Što vrijedi za procese koji se izvode paralelno u odnosu na dretve koje se izvode paralelno ?

Komunikacija među dretvama je brža nego među procesima.

**22.** Komunikacija među dretvama i procesima.

Procesi ne mogu komunicirati preko globalnih varijabli,globalne varijable su dio samo tog procesa(djeca dobiju kopiju), vec preko zajednicke memorije...Za razliku od toga dretve dijele resurse istog procesa i stoga mogu komunicirati preko globalnih varijabli