

1. Zadatak

Objašnjenje situacije

Naš glavni producent dr.sc. Aniv Vićrajko odlučio je upotrijebiti znanje filmske montaže i montirati film iz prethodne vježbe. U tome mu pomaže Lanmi Jićba koji mu sugerira da je nužno potrebno zbog rezanja i ljepljenja filmske trake utvrditi koja scena (sličica) će biti zadržana, a koja će biti odbačena. U svrhu moderne umjetnosti predlaže da umjetna inteligencija u računalu odredi koja sličica će biti zadržana. U procesu direktne pogodbe naš Prodekan mr.sc. Inkumar Garža sa Vama sklapa direktну pogodbu da izradite programsku podršku koja evidentira sličice koje će izbaciti sa brojem -1, a sličice koje će zadržati sa brojem 1. Nije bitno da program spremi podatke, već će se koristiti za vrijeme rada. Bitno je samo da u danom trenutku može primiti unose za 100000 sličica koliko u jednom mahu mogu naši montažeri izrezati i zalistiti. Naš producent i montažer inzistira da navedeni program (exe) mora stati na što manji USB štap (stick) jer će se montaža obavljati na više lokacija: Vrbik, FSB 207, Konavoska i Holjevčeva. Postavlja Vam pitanje da li je to izvedivo? Vi ste jako zabrinuti jer ako deklarirate polje od 100000 bytova u programu onda bi trebao exe biti veličine barem 100000 bytova. Da li je to točno?

Priprema (Rad kod kuće)

Student kod kuće mora riješiti zadatak.

Napisati program koji ispisuje -1 ili 1 slučajno na ekran. Program neka koristi jedno polje char tipa veličine 100000 znakova. Polje deklarirati globalno izvan tijela glavne (main()) funkcije. Program neka prvo to polje popuni brojem -1 ili 1, a nakon toga ispiše to polje na ekran (znakove iz polja). Prevesti navedeni program (compile). Sa naredbom 'ls -al' utvrditi veličinu izvršne datoteke.

```
$ ls -la
total 28
drwxr-xr-x  2 jakov jakov  4096 Nov  8 21:04 .
drwxr-x--- 13 jakov jakov  4096 Nov  8 20:47 ..
-rw-rxr-xr-x  1 jakov jakov 16224 Nov  8 21:04 random
-rw-r--r--  1 jakov jakov    367 Nov  8 21:04 random.c
```

U programu smo koristili polje od 100000 znakova veličine 1byte, zašto je izvršna datoteka manja od 100000 byte?

Izvršna datoteka je manja od 100000 bytes jer program rezervira toliko memorije u stogu prilikom izvršavanja programa, ne čita s diska.

Kada proces dohvata memoriju potrebnu za izvršavanje. Pogledajte naredbom top koliko Vaš program troši sistemskih resursa nakon pokretanja. Ako treba upotrijebite naredbu sleep na kraju programa da bi ga mogli pratiti u naredbi top.

```
PID USER      PR NI   VIRT   RES   SHR   S %CPU %MEM TIME+ COMMAND
391 jakov    20 0   3708  1920 1792 S 0.0  0.0  0:00.01 random
```

Program troši 1920KB memorije nakon izvršavanja.

2. Zadatak

Objašnjenje situacije

Na autocesti broj A1 Špičkovina-Garešnica nalaze se ulazne rampe na lokaciji Špičkovina. Svaka ulazna rampa opremljena je modernim operacijskim sustavom koji omogućava podizanje višestrukih procesa. Prilikom izrade programa uvaženi stručnjak programer nažalost nama sada nepoznat postavio je arhitekturu sustava da postoji proces roditelj koji poziva i stvara proces dijete. Za svaki novi auto koji priđe naplatnim kućicama proces dijete obavlja funkciju izdavanja kartice korisniku koji se približio automobilom. Proces roditelj koji se izvrši nakon djeteta podiže rampu. Obzirom da se koristio linux sa starijom verzijom jezgri izvršavalo se prvo dijete pa roditelj te je sve bilo u redu. Jednoga dana naš serviser gosp. Bekha došao je i zamijenio ugrađeni sustav na lokaciji sa novijom verzijom modernog operacijskog sustava. Od tada počinje problemi. Korisnik autoceste Aniv Arces sa svojom Toyotom prolazi bez problema. Nažalost nakon toga dva vlasnika najvećeg Bavarskog čuda Nit Bergerkram i Rango Čićma odlaze kao kabrioleti obzirom da je rampa od materijala koji nekoliko puta nadjačava Bavarski lim. Vlasnik Opela Icavi Digdo ulijeće nakon Bavarskih čudesa i nema nikakvih problema, iza njega slijedi vlasnica Forda inače bivša vozačica čiji je suprug vlasnik Toyote Vai Bićre. Seat vlasnika Slavamira Mićsla zaustavlja se tik do rampe u strahu međutim rampa se podiže. Vlasnici Bavarskih čuda javili su se koncesionaru u nevjericu da netko sabotira to remek djelo Njemačke industrije. Nevjerojatna činjenica koja je prelila času koncesionaru je Nit Bergerkram koji uspješno prolazi sa automobilom Alpha dok je njegovo Bavarsko čudo bilo na redovito neplaniranom servisu. Koncesionar Vas je unajmio da utvrdite zašto na sustavu samo stradavaju samo vrhunci Bavarske industrije. Vaš zadatak je utvrditi prilikom pokretanja procesa djeteta od strane procesa roditelja u koliko slučaja se javljaju anomalije u smislu pokretanja djeteta prije roditelja ili roditelja prije djeteta u odnosu na očekivano ponašanje kako bi opravdali nelogično ponašanje sustava.

Priprema (Rad kod kuće)

Student kod kuće mora riješiti zadatke.

Napisati program koji naredbom fork() stvara nove procese djecu. Proces roditelj u programu mora stvoriti 10, 100, 1000, 10000 i 100000 procesa djece. Neka se broj procesa djece unosi kao parametar komandne linije prilikom poziva programa. Proces dijete u programu ispisuje niz „Dijete #n“ gdje je n broj procesa djeteta (indeks petlje), te nakon toga završava sa radom. Proces roditelj u nastavku rada prije stvaranja novog procesa djeteta (unutar for petlje) ispisuje „Roditelj #n“ gdje je n broj procesa djeteta (indeks petlje). Prilikom poziva programa preusmjerite sadržaj u vanjsku datoteku koristeći ljudsku linuxa. Obavezno na početku programa pozovite **setbuf(stdout, NULL);**; čime sprječavate da ljudska koristi međuspremnik.

./Izvrsi 10000 > ime_vanjske.txt

Iskoristite datoteke iz zadatka i izračunajte postotak neočekivanih događaja.

Na novijim jezgrama očekivano je da će se prvo pokrenuti roditelj, a tek onda dijete. Na starijim jezgrama je obrnuto. Morate utvrditi kako se ponaša Vaš linux, a zatim odrediti postotak neočekivanog ponašanja. Kroz varijablu /proc/sys/kernel/sched_child_runs_first moguće je utjecati na redoslijed međutim vrlo nepouzdano. Na našim virtualnim mašinama je postavljena vrijednost 0. Ponoviti mjerjenja nekoliko puta (dvadesetak) da dobijete osjećaj o odstupanjima.

Roditelj #1	Dijete #1	Roditelj #1
Dijete #1	Roditelj #1	Dijete #1
Roditelj #2	Dijete #2	Roditelj #2
Dijete #2	Roditelj #2	Dijete #2
Roditelj #3	Dijete #3	Roditelj #3
Dijete #3	Roditelj #3	Roditelj #4
Roditelj #4	Dijete #4	D3
Dijete #4	Roditelj #4	Dijete #4
<i>Prvo roditelj</i>	<i>prvo dijete</i>	<i>Prvo roditelj</i>
<i>Onda dijete</i>	<i>onda roditelj</i>	<i>pa dijete</i>
		<i>Sa 1 odstupanjem od 4 =25%</i>

Možete koristi bilo koju distribuciju koju ste sami instalirali ili možete skinuti gotov image Debian 9 sa stranica linux.tuzv.hr te ga pokrenuti u vmware virtualnom okruženju (vmware workstation player je besplatan).

Za pripremu je potrebno donijeti:

- Programe na digitalnom mediju (USB) ili na oblaku (one drive, google disk...) (trebati će ih prebaciti na virtualne mašine sa kojih ćete predavati)
- Veličinu izvršne datoteke iz zadatka 1
- Graf koji je nastao temeljem 2 zadatka. Na osi X nalazi se broj pogrešaka, a na osi y mjerena za pojedine iznose brojeva procesa djece (10,100,1000,10000,100000). U grafu prikazati najveća odstupanja.

Priprema treba biti napisana na računalu ili čitko rukom. Za postojanje pripreme ne dobivaju se bodovi ali je ona uvjet za pristup vježbi.

Rad na vježbi

Student na vježbi mora obraniti zadatak. To znači da mora razumjeti na koji način je riješio zadatak iz pripreme (dva programa i rješenje za stvaranje datoteke). Uz poznавање rješenja koje je sam napravio mora razumjeti pozadinu koncepta. Usmenom obranom student može postići maksimalan broj bodova prema tablici predaje iz predavanja pravila predmeta.

Kod za programe nalazi se na GitHubu: <https://github.com/jacob6707/os-lab2>

Success rate vs. Broj djece

