**Priprema (Rad kod kuće)**

Student kod kuće mora riješiti zadatak.

1. Zadatak:

Potrebno je napisati program koji stvara procese. Broj procesa koji će biti pokrenuti unosi se preko komandne linije. U programu roditelju neka postoji globalno deklarirana varijabla *zbroj* na početku postavljena na vrijednost 0. Svaki proces dijete neka varijablu var uveća za vrijednost 1 unutar koda djeteta. Ispisati varijablu *zbroj* na kraju procesa roditelja. Zašto iznosi toliko?

1. Zadatak:

Potrebno je napisati program koji stvara dretve. Broj dretvi koje će biti pokrenute unosi se preko komandne linije. U kodu roditelj-dretve neka postoji globalno deklarirana varijabla *zbroj* na početku postavljena na vrijednost 0. Svaka dretva dijete neka varijablu *zbroj* uveća za vrijednost 1 unutar koda dretve. Ispisati varijablu *zbroj* na kraju kada sve dretve završe. Zašto iznosi toliko?

1. Zadatak

Pokrenuti na računalu zadatak1 i zadatak2 za različite brojeve procesa/dretvi za 1/10/100/1000/10000/100000. Nacrtati na istom grafu rezultate za zadatak1 i za zadatak2. Na grafu je na Y osi iznos dretvi, a na X osi izmjereno vrijeme trajanja pokretanja pojedinog zadatka. Vrijeme mjeriti kao u prvom labosu. Prikazati zasebne grafove za stvarno trajanje, vrijeme korisničkog moda i sistemsko vrijeme.

1. Zadatak

Dvoje vaših nastavnika odlučilo je low budget kompanijom otputovati na Erasmus na Kanarsko otočje. U avionu ima 100 stajaćih mjesta. Low budget kompanija nema novaca za aerodromske pristojbe te je odlučila cijeli check in proces provesti online. Za tu svrhu unajmili su putem platforme „toptal“ vrhunskog C# programera i Android Mobile Application Developera.

Programeri su predložili slijedeći algoritam:

* + osoba se prijavi na sustav za check in koristeći E-ticket broj i broj putovnice kao lozinku
    - navedena aktivnost traje između 1-5 sekundi jer su sustav riješili kroz Microsoft oblak na kojemu nije uvijek lijepo vrijeme pa sukladno se mijenja i vrijeme dohvata. U programu simulirati komandom sleep.
  + nakon spajanja na sustav dohvati polje od 100 elementa koje je zapisano na disketi od 51/4 kapaciteta 110kb koja je ubačena u disketar na farmi poslužitelja položenoj u Atlanski ocean na dubini od 10m
    - simuliramo globalnim poljem u našem programu. Dohvat realizirati posebnom funkcijom kod koje se uzrokuje kašnjenje između 1 do 3 sekunde slučajnim odabirom.
  + Korisniku se prikazuje slika aviona i korisnik izabire stajaće mjesto
    - Realiziramo u programu slučajnim odabirom stajaćeg mjesta (0-99 ili 1-100). Nije potrebno implementirati prikaz slike aviona.
  + Pita se korisnika da li je siguran da želi baš to stajaće mjesto (obavezno se prikazuje slobodna površina stajaćeg mjesta te visina pripadne ugodne šipke za pridržavanje tijekom vožnje (za detalje pogledati Grunfov avion u epizodi Alan Forda).
    - Simuliramo sa slučajnim odabirom. U 50% slučaja korisnik odbija mjesto i ponavlja se proces biranja. U 50% slučaja izabire se to sjedalo. Ako korisnik odbije sjedalo treba mu ponuditi alternativno dok ga ne prihvati.
  + Preko API-a u oblaku se zapisuje promjena navedenog sjedala na zauzeto
    - Procedura traje od 1-2 sekundi slučajnim odabirom. Potrebno je samo zapisati u odgovarajući element polja broj 1 (zauzeto).

Avionska kompanija je pristala na algoritam i programeri su kratkom roku implementirali navedeni sustav. Naplatili su ga $10000 te u Beta verziji pokrenuli za potrebe ukrcaja na let na koji idu vaša dva cijenjena nastavnika. Na aerodromu se stvorila gužva kada je pomoćnica čistačice sanitarnih prostora aerodroma (ovlaštena djelatnica low budget kompanije) putem sofisticirane tehnologije (savinutog papira u obliku megafona) zamolila putnike da putem svojih mobilnih uređaja pokrenu check-in. Nažalost nisu svi putnici bili spojeni na mobilnu mrežu u tome trenutku tako da je slučajan broj njih krenuo odmah, a ostatak sekundu kasnije (simulirati kroz slučajan broj 0-99 ili 1-100).

Upute za realizaciju:

Stvoriti polje od 100 elemenata (10x10). Stvoriti 100 dretvi (korisnika). Svaka dretva se pokreće u zadano vrijeme (kašnjenje zbog spajanja na WIFI) prema algoritmu odrađuje svoju aktivnost i završava sa radom. Program mora ispisivati konačni broj zauzetih i slobodnih sjedala.

* Pokrenuti program više puta i analizirati različite dobivene vrijednosti.
* Razmisliti zašto ne valja algoritam programera?
* Razmisliti o mogućem rješenju problema.

Možete koristi bilo koju distribuciju koju ste sami instalirali ili možete skinuti gotov image Debian 9 sa stranica linux.tvz.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)
* Ispis primjera rada programa

***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 poznavanje rješenja koje je sam napravio mora razumjeti pozadinu koncepta. Usmenom obranom student može postići maksimalan broj bodova prema tablici predaje iz predavanja pravila predmeta.