

Rešenje ispita 21.07.2022.

Napomena: Da bi bili bodovani nije neophodno da vaši odgovori budu apsolutno identični ovima navedenim ovde. Predstavljaju manje-više minimalnu formu za maksimalan broj bodova.

Grupa A

1. Stranični, Segmentni, i Kontinualni
2. Preplitanje može uticati na rezultat izvršavanja programa u formi *štetnog preplitanja*. Ovo se dešava kada više niti pristupa nekakvoj deljenoj promenljivoj i, zbog nepredvidive stohastičke prirode preključivanja između niti, redosled pristupanja varira između izvršavanja programa i ponekad se desi namenjenim redosledom, a ponekad se pristupi preklape tako da jedna nit vidi deljenu promenljivu u nekonzistentnom stanju koje, kada se modifikuje, dovodi do promene rezultata i grešaka.
3. Ne bih koristio fajl sistem sa pregledom izmena odn. Journaling mehanizmom. Glavni razlog za to jeste što, prvo, ne očekuje se da tih izmena uopšte i bude, budući da se sadržaj diska piše jednom, a drugo čak i da se desi nekakva situacija gde bi praćenje izmena bilo relevantno, sa tim praćenjem ne može ništa da se uradi budući da se sadržaj diska, ako je upisan u nekonzistentnoj formi prilikom svog jedinog pisanja, ne može dalje menjati¹.
4. Morao bih da primenim nekoliko mera da ublažim posledice tako loše mrežne veze. Među tim merama je: paketizacija, retransmisija, kontrolne sume, i striktno idempotentan protokol za komunikaciju. Paketizacija omogućava da se poruka polako rekonstruiše iz manjih delova, ako je poruka velika. Retransmisija šalje i poruku i njen odgovor više puta: dok god samo jedna vrednost prođe, komunikacija se uspostavila. Statistički govoreći, može se postići verovatnoća od otprilike 99% barem jednog uspešnog slanja tako što se vrši 10 slanja. Kontrolne sume kao što su md5 ili SHA256 ili bitovi parnosti garantuju da poruka nije oštećena u putu. Idempotencija je jako bitna i sprečava da se retransmisijom možda pokrenu neke operacije više puta kada to nije bilo namenjeno. Idempotentnost se postiže numerisanjem poruka i jasnim označavanjem retransmisionih kopija. Rizici su nedostatak pouzdanosti, i neželjeno višestruko izvršavanje zbog nedostatka idempotentnosti.

Grupa B

1. Smanjuje se stepen iskorišćenja procesora, dobro je za multimedijalnu upotrebu, ubrzava se odziv.
2. Svaka nit konkurentnog procesa ima svoje stanje (koje može biti aktivna, spremna i čeka, baš kao i stanja procesa), svoj prioritet, svoj stek, i svoju aktivnost, pa prirodno i svoj deskriptor gde neki su smešteni neki od ovih podataka. Ovi podaci se delimično kreiraju kada se kreira i nit, a delimično su nasleđeni od procesa. Sve niti jednog

¹ Radi jednostavnosti a i zbog toga što to na predmetu nismo pominjali, ovaj odgovor ignoriše određene egzotične detalje vezane za višestruko "narezivanje" i lead-in i lead-out staze i njihovu interakciju sa konzistentnošću materijala.

konkurentnog procesa dele adresni prostor i celu sliku procesa (što uključuje i memoriju gde su promenljive i mašinski kod) osim steka.

3. Postoji velika pretnja od instaliranja malicioznih drajvera i potencijalna šteta ako bi se na računar instalirao maliciozan drajver je znatno veća nego kod instaliranja malicioznog aplikativnog programa, ili posete malicioznom sajtu (iako su sve ovo stvari koje definitivno treba izbeći). Razlog za veliku štetu od malicioznog drajvera jeste što se kod drajvera izvršava unutar operativnog sistema u sistemskom adresnom prostoru. To daje drajverima apsolutan pristup sadržaju operativnog sistema koji se ne može zaštititi jer je sam mehanizam za bezbednost deo operativnog sistema koji je sada infiltriran.
4. Postoji razlika između programskog jezika koji je dizajniran da bude konkurentan i programskog jezika kome je konkurentna biblioteka deo standardne biblioteke. Oba jezika podržavaju konkurentnost bez ikakvih daljih modifikacija. Glavna razlika je u tome što konkurentan jezik ima strukture unutar samog jezika koje menjaju njegovu sintaksu i koje služe da implementiraju mehanime vezane za konkurentnost i sinhronizaciju. U slučaju jezika koji to ima u biblioteci, on zadržava standardnu, sekvencijalnu sintaksu koju proširuje funkcijama i klasama koje pružaju usluge konkurencije i sinhronizacije. Konkurentni jezici mogu biti prilagođeniji problemu i proizvoditi čistiji kod, a konkurentne biblioteke omogućavaju da se koriste dokazani jezici koji se mogu primeniti i na druge oblasti, po potrebi.