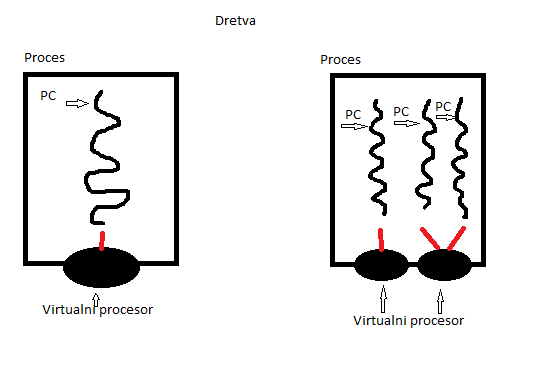
1. Koja je razlika izmedu procesa nekada i danas?

* Nekada: Danas:

1. Koja je razlika između jednoprocesorske obrade i višeprocesorske obrade?

* Jednoprocesorsko: prividno više radnji simultano
* Višeprocesorsko: simultano više radnji

1. Što imaju procesi zajedničko?

* Ništa (adresni prostor je podijeljen)

1. Što je zajedničko dretvama istog procesa?

* Sve

1. Gdje se nalaze globalne varijable?

* U zajedničkom adresnom prostoru.

1. Koliko minimalno dretvi mora sadržavat proces?

* Proces se sastoji od barem jedne dretve

1. Kako se naziva novi proces ili više njih koji stvara proces roditelj?

* Proces dijete

1. Što možemo reć za proces dijete?

* Proces dijete je kopija procesa roditelj

1. Da li su podatci zajednički procesu roditelj i dijete?

* Nisu

1. Da li proces roditelj mora nadživjeti proces dijete?

* Da

1. Da li svaki proces mora imati svojeg roditelja?

* Da

1. Ako ubijemo roditelja što se dešava s tekućim procesom dijete?

* Proces dijete postaje ORPHAN i os postaje roditelj tog procesa.

1. Što radi naredba int fork(void)?

* Račva proces

1. Koje su vrijednosti int fork(void)?

* fork() takes no argument, and returns a process ID (PID)
* but after fork() has been called there are two different processes running--the parent process, which called fork(), and the child process.
* We say that fork() is called once, but returns twice.
* The parent process is returned the process ID of the newly spawned child, and the child is returned the value 0.
* By checking the return value, a process can determine whether it is the parent or child process. fork() returns -1 on error, and creates no child

1. Koji proces nazivamo ZOMBIE?

* Proces koji je završio izvođenje ali u tablicama os-a postoji njegov zapis

1. Od čega se satoji prekidni sklop?

* Tablica adresa rutine za obradu podataka
* Tekući prioritet
* Kontrolne zastavice
* Prepoznavanje prioriteta

1. Koje žice idu prema procesoru od sklopa za prihvat prekida?

* Talica adresa
* Žica prekida

1. DMA ima posebni pristupni sklop koji prema procesoru trazi 2 stvari, koje su to?

* Sabirnicu
* Prekid

1. Koja žica se vraća prema DMA?

* Zovi sabirnicu

1. Koje radnje mogu izazvati prekide?

* Djeljenje s 0
* Deklaracija nepostojece naredbe
* Adresa lokacije koja ne pripada CPU

1. Kada se događa ulazak u jezgru?

* Kod prekida

1. Što imaju zajedničko procesi/2 dretve?

* Ništa/sve

1. Kako se zovu procesi?

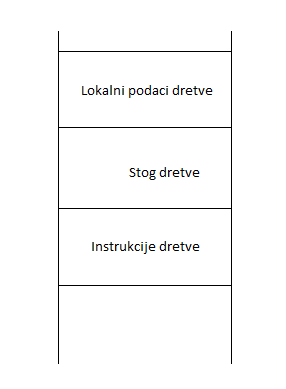
* PID

1. Kako se zovu dretve?

* TID

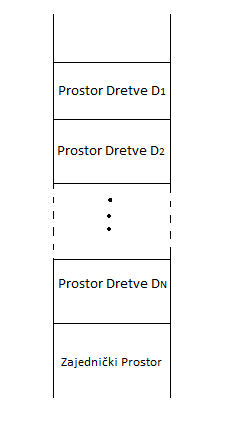
1. Što je proces dijete?

* Kopija roditelja

1. Od čega se satoji drevtveni dio spremničkog prostora

* Lokalni podaci
* Stog dretve
* Instrukcije dretve

1. Kako izgleda podjela procesnog spremničkog prostora?

* Prostor za N dretvi
* Zajednički prostor

1. Kako se naziva podskup zajedničkog spreminka s kojeg dretva Di započinje svoje izvođenje „čita“?

* Domena Xi

1. Kako se naziva podskup zajedničkog spreminka u kojeg dretva Di zapisuje rezultate?

* Kodomena Yj

1. Po opisanom čitanju iz Domene u lokalne varijable i pisanju u Kodomenu rezultate, čemu je dretva slična?

* Potprogramu

1. Kako bi prikazali izvođenje dretve pomoću funkcije'

* fi :Xi → Yj
* Yj = fi (Xi)

1. Kako nazivamo 2 zadatka koji nemaju zajedničkih lokacija u svojim domenama i kodomenama?

* Međusobno nezavisni

1. Da li se međusobno nezavisni podzadaci mogu izvoditi proizvoljnim redosljedom?

* Da

1. U kojem slučaju međusobno nezavisni zadatci mogu imati istu domenu?

* U slučaju kada te lokacije služe samo za čitanje ulaznih podataka

1. Izrazi uvjet nezavisnosti:

* (Xi ∩ Yj) U (Xj ∩ Yi) U (Yi ∩ Yj) = ∅

1. Koji je drugi naziv za sustav podzadataka?

* Sustav dretvi

1. Kako je najlakše predočiti sustav dretvi?

* Usmjerenim grafom

1. Što predstavljaju čvorovi usmjerenog grafa a što strelice?

* Čvorovi: pojedine dretve
* Strelice: redosljed izvođenja dretvi

1. Kako izgledaju i kako se izvode međusobno zavisne dretve u usmjerenom grafu?

* To su dretve koje se nalaze na istom putu od početne do završne dretve te se moraju izvoditi propisanim redosljedom

1. Kako izgledaju i kako se izvode međusobno nezavisne dretve u usmjerenom grafu?

* To su dretve koje nisu na istom putu i mogu se izvoditi proizoljnim redosljedom

1. Što znači da se dretve mogu izvoditi paralelno?

* To znači da se dretve nalaze na paralelnim putevima

1. Kako mogu biti izvođene paralelne dretve?

* Istodobno ako za njihovo izvođenje imamo na raspolaganju vise fizičkih procesora
* Prvidno istodobno ako se one izvode na jednom procesoru

1. Što je sinkronizacijski mehanizam?

* Omogućuje pokretanje i zasutavljanje dretvi, te načini na koji jedna dretva obavjesti drugu da je ona završila izvođenje

1. Koje su mogučnosti razmjene podataka kod dretvi?

* Jedna dretva piše u domenu druge dretve
* Pomoću poruka koje moraju imati dogovoreni oblik, a razmjenjuju se posebnim, rezerviranim djelom zajedničkog spremnika

1. Što omogučava mehanizam međusobnog isključivanja?

* Osigurava da se neka sredstva računalnog sustava koriste pojedinačno.

1. Što su kritični odsječci?

* Dijelovi dretvi koji koriste neko zajedničko sredstvo

1. Kako nezavisne dretve prolaze kroz svoj K.O?

* Smiju prolaziti kroz svoj K.O. samo pojedinačno

1. Što su zapravo Kritični odsječci?

* To su neprekidivi dijelovi obrade prekida

1. Kako ostvariti međusobno isključivanje na jednoprocesorskom računalu?

* Zabranom prekidanja

1. Napiši psudokod Cikličke dretve:

Dok (1){

K.O;

N.K.O;

}

1. Koji su uvjeti koje mora zadovoljiti algoritam M.I?

- Dretve izvode K.O. međusobno isključivo

- Mehanizam M.I. mora djelovati i onda kada su brzine izvođenja proizvoljne

- Kada neka dretva zastane u N.K.O. to ne smije spriječiti drugu dretvu da uđe u K.O.

Izbor koja dretva ulazi u K.O. mora se odviti u konačnom vremenu

1. Zašto prvi pokušaj ostvarivanja M.I. ne valja i koliko varijabli uvodimo?

* Jedna varijabla ZASTAVICA
* Može se dogoditi da obje zastavice istovremeno žele ući u kritični odsječak i da one obje pročitaju istu vrijednost varijable ZASTAVICA, tj. Nađu 0. To se može dogoditi ako su dretve narazličitim procesorima te u uzastopnim sabirničkim ciklusima pristupaju toj varijabli
* Može se desiti i na jednoprocesorskim sustavima ako se prvu dretvu koja je upravo pročitala zastavicu prekine prekidom nakon te instrukcije.

1. Zašto drugi pokušaj ostvarivanja M.I. ne valja i koliko varijabli uvodimo?

* Imamo dvije varijable ZASTAVICA i PRAVO
* Ovaj pokušaj rješenja omogučuje, da se dretve odvijaju u svojim K.O. samo naizmjence. Jedna od dretvi ne može dva puta uzastopce ući u K.O.

1. Zašto treći pokušaj ostvarivanja M.I. ne valja i koliko varijabli uvodimo?

* Imamo 2 varijable ZASTAVICA[i] i ZASTAVICA[j]
* Obje varijable mogu pokušati ući istovremeno u K.O. One pritom mogu obje pročitati u 2 uzastopna sabirnička ciklusa vrijednosti 0, svaka iz zastavice suparničke dretve, i na taj način mogu obje uči u K.O.

1. Zašto četvrti pokušaj ostvarivanja M.I. ne valja i koliko varijabli uvodimo?

* Imamo 2 varijable ZASTAVICA[i] i ZASTAVICA[j]
* Tu se pojavljuje pojava dead lock, kada obje dretve istovremeno zažele ući u K.O., i u dva uzastopna sabirnička ciklusa se može dogoditi da se u zastavice upiše 1. Nakon toga jedna i druga ulaze svaka u sovju petlju čekalicu i tamo ostaju trajno.

1. Zašto peti pokušaj ostvarivanja M.I. ne valja i koliko varijabli uvodimo?

* Imamo 2 varijable ZASTAVICA[i] i ZASTAVICA[j]
* Pročitati u knjizi str. 76.

1. Koja je kombinacija rješenje ovog problema?

* Kombinacija drugog i petog pokušaja

1. Koliko imamo varijabli u Dekkerovom postupku?

* Imamo 2 varijable ZASTAVICA[i] i ZASTAVICA[j]
* PRAVO

1. Koliko imamo varijabli u Petersonovom postupku?

* Imamo 2 varijable ZASTAVICA[i] i ZASTAVICA[j]
* NE\_PRAVO

1. Koji je pojednostavljenje kojeg postupka?

* Petersonovom je pojednostavljeni Dekkerov postupak

1. Koje surazlike iznmeđu ta 2 postupka?

* Utjecaj na varijablu PRAVO
* Deker daje prednost sporijoj dretvi
* Petersonov ima manje koda, manje memorije trosi

1. Koja je razlika između Lamportovog protokola i navedenih?

* Prozvao ga je pekarskim algoritmom
* Koristimo ga za N dretvi a prethodno navedene za 2 dretve