

### Koncept procesa

U prvobitnim računarskim sustavima izvodili su se programi slijedno jedan za drugim. Program koji se izvodio imao je potpun nadzor nad računarskim sustavom, odnosno pravo uporabe svih njegovih resursa. Današnji računarski sustavi dozvoljavaju da se više programa unese u memoriju računala i da se istovremeno izvode. Ovakav pristup zahtjeva složeniji sustav upravljanja i nadzora nad resursima računala i praćenje izvođenja pojedinih programa. Ovi zahtjevi rezultirali su uvođenjem pojma procesa koji se odnosi na program koji je u izvođenju. Proces postaje jedinica obrade u suvremenim računarskim sustavima s vremenskom podjelom poslova.

Složeniji operacijski sustavi pružaju znatno veću podršku korisniku. Iako je težište na izvođenju korisničkih programa, potrebno je voditi računa i o različitim sistemskim zadacima koji su izvan jezgre operacijskog sustava. Tako se sustav sastoji od skupine procesa: korisničkih procesa koji izvode korisnički kod i procesa operacijskog sustava koji izvode sistemski kod. Ovi procesi izvode se paralelno dijeleći međusobno procesor. Prebacivanjem izvođenja s procesa na proces moguće je postići značajno veću učinkovitost računarskog sustava.

Proces je, kao što je već napomenuto, program u izvođenju. Procesor izvodi proces izvodeći slijedno naredbu po naredbu procesa. Proces je znatno više od programskog koda koji se izvodi. On obuhvaća i sve trenutne aktivnosti u sustavu opisane sadržajima spremnika procesora i sadržajima memorijskih lokacija koje koristi proces. Tako proces općenito sadrži programski odsječak ili nepromljenjivi tekstualni segment, stog procesa koji sadrži privremene podatke potrebne procesu (podatke koje se prenose u podprogram, povratnu adresu iz podprograma i trenutne varijable) i globalne podatke pohranjene u dio podatkovne memorije (data section).

Važno je napomenuti da program sam za sebe nije proces. On je pasivna struktura pohranjena kao datoteka u sekundarnoj memoriji (disku). Proces je aktivni entitet, odnosno program u izvođenju s programskim brojiлом koje pokazuje na sljedeću naredbu procesa i pripadajućim skupom spremnika i memorijskih lokacija kao i resursa računarskog sustava koje koristi.

vektor spremnika  $r = \{r_1, r_2, \dots, r_i\}$

vektor memorije  $m = \{m_1, m_2, \dots, m_j\}$

Skup  $S = \{r, m\}$  opisuje u svakom trenutku stanje procesa. Proces izvođenje započinje s nekim početnim stanjem  $S_0 = \{r_0, m_0\}$  koji sadrži ulazne podatke, propagira se kroz niz stanja prema konačnom stanju  $S_n = \{r_n, m_n\}$  koji sadrži izlazne podatke.

$S_0, S_1, S_2, \dots, S_k, S_{k+1}, \dots, S_{n-1}, S_n$

Standardni tipovi računala imaju svojstvo da je stanje  $k+1$  trenutku funkcija stanja u  $k$ -tom:

$S_{k+1} = f_k(S_k)$ .

Tako je za svaki proces unaprijed definiran i niz funkcija:

$f_0, f_1, f_2, \dots, f_k, f_{k+1}, \dots, f_{n-1}$

Ovo svojstvo omogućava da se proces može u svakom trenutku prekinuti i nakon toga nastaviti ukoliko se sačuva stanje procesa u trenutku prekida  $S_k$ . Prebacivanje izvođenja s procesa na proces koje ima za posljedicu promjenu sadržaja spremnika procesora naziva se promjena konteksta. Važno je ustanoviti uvjete kada je neki proces moguće prekinuti, te kada ga je moguće ispravno nastaviti.

Problem je odrediti kada je moguće neki proces  $p_i$  prekinuti i zamijeniti ga procesom  $p_j$ , s mogućnošću da se proces  $p_i$  naknadno nesmetano nastavi. Svaki proces tijekom izvođenja koristi određene spremnike i određene memorijske lokacije. Tako neka:

**----- OSTATAK TEKSTA NIJE PRIKAZAN. CEO RAD MOŽETE  
PREUZETI NA SAJTU. -----**

[www.maturskiradovi.net](http://www.maturskiradovi.net)

MOŽETE NAS KONTAKTIRATI NA E-MAIL: [maturskiradovi.net@gmail.com](mailto:maturskiradovi.net@gmail.com)