

PROCESI I DRETVE

1. Proces

- osnovna jedinica rada koju O.S. koristi za izvršavanje programa
- predstavlja pokrenutu instancu programa, te sadržava sve potrebne resurse za izvršavanje, uključujući memoriju, registre, *stog*, i upravljačke podatke.

1.1 Ključne karakteristike procesa:

A) Adresni prostor:

Proces ima svoj vlastiti VAP (1), što znači da mu je dodijeljena memorija koju ne dijeli s drugim procesima. Ovo nam omogućuje sigurnost i stabilnost, jer sprječava da jedan proces utječe na podatke drugoga.

B) Izolacija resursa:

Procesi su međusobno izolirani, tako da jedan proces ne može pristupiti memorijskim resursima ili podacima drugog procesa bez posebnih dozvola.

C) Identifikacija:

Svaki proces ima jedinstveni identifikator, poznat kao *PID*(2), koji mu omogućuje identifikaciju unutar O.S.

1 - virtualni adresni prostor

2 – *process Identifier*

1.2 Stanja procesa:

Proces može biti u jednom od nekoliko stanja.

<i>New</i>	Proces se kreira, ali još nije spreman za izvršavanje.
<i>Ready</i>	Proces čeka na CPU kako bi započeo izvršavanje.
<i>Running</i>	Procesor trenutno izvršava proces.
<i>Blocked</i>	Proces čeka na neki resurs ili događaj (npr. ulazno-izlaznu operaciju).
<i>Exit</i>	Proces je završio izvršavanje.

1.3 *Process Control Block (PCB)*:

O.S. koristi PCB za upravljanje procesima. PCB pohranjuje sve ključne informacije o procesu, uključujući:

- Identifikator (PID)
- Stanje procesa
- Programski brojač (adresa sljedeće instrukcije)
- Memorijske pokazivače
- Informacije o korištenju resursa

1.4 Primjer korištenja procesa:

Kada korisnik otvori 2 različita programa (poput preglednika i tekstualnog editora), svaki od tih programa izvršava se u zasebnom procesu s vlastitim resursima, čime se omogućava izolacija, a O.S. upravlja njihovim izvođenjem kako bi oba programa mogla raditi paralelno, bez međusobnih smetnji.

2. Dretva (thread)

“Laka verzija procesa”, odnosno **jedinica izvršavanja unutar procesa**. Dretve omogućuju **paralelno izvršavanje** različitih dijelova programa unutar **istog procesa**. **Svaka dretva dijeli memoriju i resurse** svog matičnog procesa, što omogućava bržu komunikaciju između dretvi u usporedbi s komunikacijom između procesa.

2.1 Ključne karakteristike dretvi

A) Dijeljeni resursi:

Dretve unutar istog procesa dijele adresni prostor, globalne varijable i resurse, što ih čini lakšima za kreiranje i održavanje od potpuno zasebnih procesa.

B) Neovisni stogovi:

Iako dijele adresni prostor, svaka dretva ima vlastiti stog, koji omogućuje pohranu lokalnih varijabli i povratnih adresa specifičnih za tu dretvu.

C) Brža komunikacija:

Zbog zajedničkog adresnog prostora, dretve mogu brže i jednostavnije razmjenjivati podatke nego procesi.

2.2 Vrste dretvi:

A) Korisničke dretve (ULT): upravljaju se na razini APK; OS nije svjestan njihovog postojanja.

B) Jezgrine dretve (KLT): upravljaju se od strane OS jezgre; omogućuju bolje iskorištavanje višestrukih procesora.

2.3 Stanja dretve:

Dretve imaju slična stanja kao i procesi:

<i>Ready</i>	Dretva čeka da dobije procesorsko vrijeme.
<i>Running</i>	Dretva se aktivno izvršava.
<i>Blocked</i>	Čeka na resurs ili događaj.
<i>Finished</i>	Završila je svoje izvršavanje.

2.4 Prednosti korištenja dretvi:

A) Veća učinkovitost:

Manje vremena potrebno je za kreiranje i prebacivanje između dretvi unutar istog procesa nego između različitih procesa.

B) Povećana brzina izvršavanja:

Dretve omogućuju paralelno izvođenje više zadataka unutar istog procesa, što povećava brzinu i responzivnost programa.

Primjer korištenja jedne dretve:

Zamislimo jednostavnu APK koja obavlja samo **jednu funkciju**, kao što je preuzimanje datoteke s interneta. U ovom slučaju, APK koristi **jednu dretvu** koja upravlja cijelim **procesom preuzimanja**. Dretva će započeti preuzimanje, pratiti napredak i završiti zadatak kad je datoteka preuzeta. Sve operacije vezane uz preuzimanje obavlja ta jedna dretva, koja ne izvodi druge zadatke dok se preuzimanje ne dovrši.

Time možemo zaključiti da je jedna dretva odgovorna za određeni dio izvršavanja programa unutar procesa APK.

3. Razlika između procesa i dretvi

Osnovna značajka	Proces	Dretva
Adresni prostor	Svaki proces ima vlasiti prostor	Dijele adresni prostor procesa
Izolacija	Izoliran od drugih procesa	Dijeli resurse s ostalim dretvama unutar istog procesa
Stvaranje i prebacivanje	Sporije	Brže
Primjer	Tekstualni editor	U tekstualnom editoru, dretva za unos teksta i dretva za automatsko spremanje rade paralelno

4. Multithreading i multitasking

Multithreading omogućuje pokretanje više dretvi unutar jednog procesa, što poboljšava paralelno izvršavanje zadataka.

Primjer: aplikacija koja istovremeno obrađuje podatke i prikazuje rezultate.

Multitasking omogućava pokretanje više procesa na sustavu.

Primjer: situacija kada istovremeno rade preglednik i editor teksta.

4.01 Prednosti višedretvenosti:

- A) **Brža komunikacija:** Dretve dijele memorijski prostor, što omogućuje bržu međudretvenu komunikaciju.
- B) **Ekonomičnost:** Kreiranje i prekidanje dretvi je brže nego kod procesa.
- C) **Učinkovitost:** Mogućnost paralelnog izvršavanja različitih dijelova programa.

4.1 Jednodretveni vs. *Multithreading* pristupi:

A) Jednodretveni:

Svaki proces ima samo jednu dretvu. Primjer je MS-DOS.

B) Multithreading:

Omogućuju više dretvi unutar jednog procesa, što poboljšava učinkovitost iskorištavanja resursa.

5. Zaključak

Korištenje procesa i dretvi optimizira učinkovitost i responzivnost programa. Proces i omogućuju izolaciju aplikacija, dok dretve unutar procesa poboljšavaju brzinu izvršavanja unutar pojedinog program

3. Kernel

Osnovni dio, to jest jezgra O.S.-a, koji osigurava pristup hardverskim resursima i upravlja procesima.

3.1 Glavne funkcije:

- A) **Upravljanje procesima:** Kernel kreira, terminira i planira izvršavanje procesa koristeći algoritme rasporeda.
- B) **Upravljanje memorijom:** dodjela adresnog prostora, zamjena stranica i segmentacija.
- C) **Upravljanje I/O uređajima:** kontrolira komunikaciju između procesa i I/O uređaja.

Kernel također osigurava sigurnost i izolaciju između procesa kako bi spriječio neželjene interferencije između njih.