

Jezgro OS i upravljanje procesima

Jezgro (kernel) osnovni je deo svakog operativnog sistema. U hijerarhiskom (slojevitom) modelu, jezgro je najbliže hardveru i predstavlja vezu tj interfejs između hardvera i ostalih slojeva operativnih sistema. To je deo operativnog sistema koji omogućava programima da na siguran način pristupe hardveru.

Proces je jedan od najvažnijih koncepata operativnih sistema. To je deo programa ili program u stanju izvršavanja, zajedno sa resursima potrebnim za rad programa.

Jezgro

Jezgro je osnovni deo svakog OS. U slojevitom modelu ono je najbliže hardveru i nadovezuje se direktno na hardver, obezbeđujući vezu između hardvera i ostalih slojeva OS. Izuzetak predstavlja operativni sistem Windows u čijoj se osnovi nalazi NT arhitektura. U slojevitom modelu NT arhitekture, ispod jezgra se nalazi sloj apstrakcije hardvera (*Hardware Abstraction Layer, HAL*) koji omogućava operativnom sistemu da vidi različit hardver na isti način.

Jezgro se može i definisati kao deo operativnog sistema koji omogućuje programima da na siguran način pristupe hardveru. Jezgro OS-ma nije neophodno za pokretanje i izvršavanje programa. Programi i procesi se mogu izvršavati na računaru bez jezgra ukoliko je programer spremna zaobiđe apstrakciju hardvera i usluge koje operativni sistem pruža i program napiše na niskom nivou, koristeći instrukcije mašinskog jezika. Ovaj način programiranja i izvršavanja programa je bio uobičajen na starijim računarskim sistemima. Nemogućnost prelaska iz jednog programa u drugi predstavljava je osnovni nedostatak ovakve realizacije – korisnik koji je želeo da prekine rad u jednom programu i pokrene drugi program, morao je da resetuje računar. To se sprečavalo malim pomoćnim programima za otkrivanje grešaka (*debuggers*) i alatkama koje su punile memoriju programskim kodom. Ovi pomoćni programi koji su omogućavali prelazak iz jednog programa u drugi bez resetovanja računara predstavljaju osnovu za dalji razvoj jezgra operativnog sistema.

Osnovna funkcija jezgra je upravljanje procesima tj stvaranje okruženja u kome mogu postojati procesi, dodeljivanje procesora procesima i obezbeđivanje mehanizama za interprocesnu komunikaciju. Kako se na procesoru u jednom trenutku može izvršavati samo jedan proces (proces je nedeljiv resurs), jezgro određuje kada i na koje vreme će proces dobiti procesor. Ova pojava je poznata pod imenom **multiplesiranje** i predstavlja osnovu **kvaziparalelnosti**.

Da bi jezgro ostavarilo svoju osnovnu funkciju, neophodno je da sam hardver ispunjava određene preduslove tj da na nivou hardvera postoje komponente koje omogućuju nadogradnju hardvera jezgrom. U te komponente spadaju:

- Mehanizam prekida
- Zaštitni mehanizam adresiranja memorije
- Skup privilegovanih instrukcija
- Časovnik realnog vremena

Mehanizam prekida

Mehanizam prekida obezbeđuje izvršenje upravljačkog programa (prekidna rutina) tj prebacivanje kontrole izvršavanja s korisničkog na upravljački program. Najmanje što mehanizam prekida treba da učini jeste da sačuva vrednosti programskog brojača prekinutog programa i pokrene upravljački program s fiksne lokacije u memoriji. Upravljački program dalje određuje izvor prekida i reaguje na odgovarajući način.

Zaštitni mehanizam adresiranja memorije

Zaštitni mehanizam adresiranja memorije sprečava pogrešno adresiranje tj onemogućava da jedan proces upiše prateće podatke u deo memorije koji je dodeljen drugom procesu. Ovaj mehanizam čuva integritet procesa i podataka koji se nalaze u radnoj memoriji.

Skup privilegovanih instrukcija

Skup privilegovanih instrukcija računara čine sve instrukcije koje su dostupne operativnom sistemu ali ne i korisničkim programima. Ove instrukcije omogućavaju operativnom sistemu da maskira prekide, dodeli procesor drugom procesu, pristupi zaštićenim registrima u memoriji, izvrši ulazno-izlazne operacije ili zaustavi procesor. Za razliku od neprivilegovanih instrukacija koje se izvršavaju u korisničkom režimu, prilikom izvršavanja privilegovanih instrukacija operativni sistem se nalazi u sistemskom režimu (*supervisory mode*). Korisnički program ne može da izvrši privilegovanu instrukciju direktno, već preko sistemskih poziva. Korisnički program pomoću sistemskog poziva (*system call*) zahteva od operativnog sistema da izvrši privilegovanu instrukciju. Nakon toga operativni sistem prelazi u sistemski režim i izvršava tu instrukciju.

Časovnik realnog vremena

Pomoću satnog mehanizma realnog vremena (*real-time clock*) kontroliše se i evidentira potrošnja resursa računara za sve pojedinačne procese. Ovaj mehanizam se može koristiti i za raspoređivanje i zakazivanje izvršenja raznih poslova.

Delovi jezgra

Jezgro teoretskog modela operativnog sistema može se podeliti na 3 osnovne celine:

- Prvi nivo obrade prekida
- Dispečer sistema (planer poslova niskog nivoa)
- Rutine za ostvarivanje interprocesne komunikacije

Prvi nivo obrade prekida

Prvi nivo obrade prekida (*First Level Interrupt Handler, FLIH*) čine rutine za određivanje izvora prekida i iniciranje servisa, tj opsluživanje pojedinih vrsta prekida. FLIH odgovara na spoljašnje prekide i sistemski pozive. Posle izvršavanja koda koji čini ovaj deo operativnog sistema, prekid se smatra opsluženim a dispečer odlučuje kom će procesu predati procesor na korišćenje

Dispečer sistema

Dispečer (*dispatcher, low-level scheduler*) deo je jezgra koji dodeljuje procesor procesima. Pri tome, procesor se uvek dodeljuje na osnovu nekog algoritma, kao što je *Shortest Job First* (prvo poslovi koji zahtevaju najmanje procesorskog vremena).

Rutine za ostvarivanje interprocesne komunikacije

Rutine za ostvarivanje interprocesne komunikacije deo su jezgra OS koji obezbeđuje komunikaciju između procesa. Postoji više načina ove komunikacije, kao što su slanje poruka (*send message*), semaforske tehnike, korišćenje imenovanih cevi (*named pipes, UNIX*), i korišćenje deljive memorije