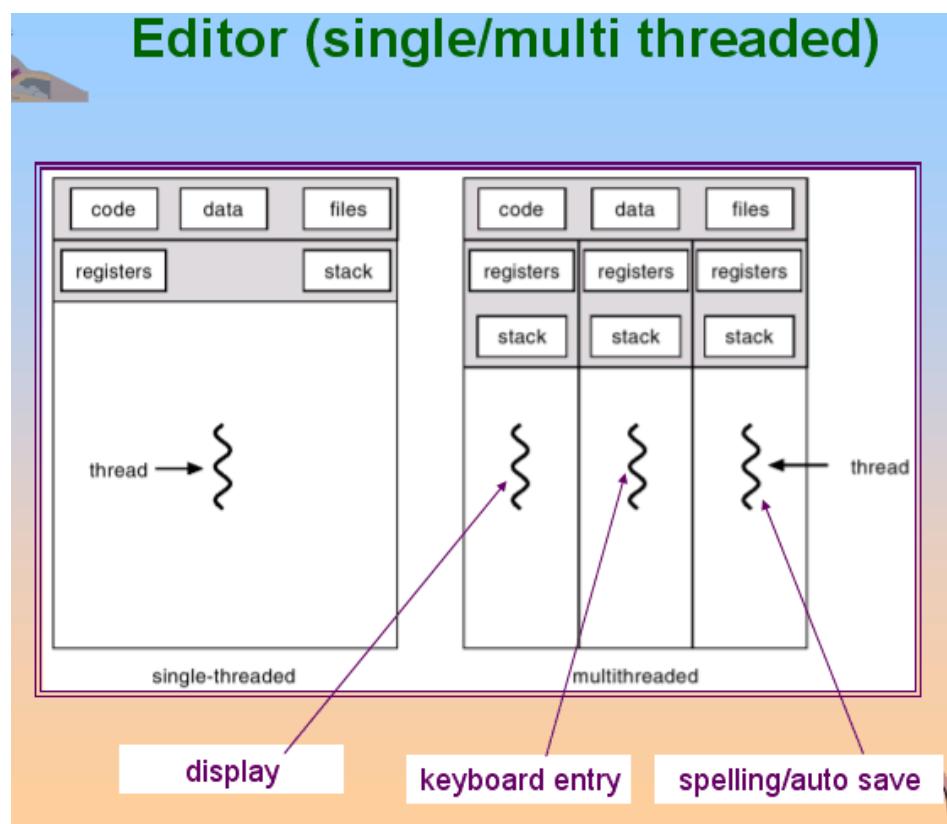


Laki i teški procesi

Niti (threads), tj laki procesi (lightweight processes, LWP), prestavljaju osnovne celine za izvršavanje koda pod savremenim operativnim sistemima. Niti (jedna ili više) pripadaju jednom klasičnom, tj teškom (heavyweight) procesu. U klasičnom kontekstu, jedan težak proces ima svoj programski brojač i druge procesorske registre, tri memorijске sekcije (kôd, podaci, stek) i ulazno-izlazne resurse kao što su datoteke i uređaji.

Niti kao laki procesi i delovi jednog istog procesa, imaju svoje unikatne resurse i zajedničke sa ostalim nitima istog procesa. Od unikatnih resursa imaju poseban identifikator niti (thread ID), posebnu vrstu programskog brojača vrednosti drugih resursa procesora i poseban stek. Sve ostalo (sekcija koda, sekcija podataka, otvorene datoteke, signali) zajednički su resursi za sve niti jednog procesa. Pomoću ovog savremenog koncepta, proces sa svojim nitima može obavljati više poslova simultano, pri čemu je prebacivanje konteksta sa jedne niti na drugu daleko brže i efikasnije od prebacivanje konteksta između teških procesa (čuvaju se i rekonstruišu samo registri procesora i stek, a sve ostalo je zajedničko).



Osim efikasnosti i veće brzine odziva, motivacija za upotrebu niti leži u zahtevima savremenih sistema zasnovanih na Internetu, gde jednom Web serveru stiže veliki broj različitih zahteva od korisnika. Mnogi moderni softverski paketi su višenitni – programi za prikazivanje Web stranica, čitači Weba (Web browser) po pravili imaju bar dve niti, od kojih jedna prikazuje hipertekst na ekranu, dok druga prima podatke sa mreže. Programi za obradu teksta imaju jednu nit za prikazivanje grafike, drugu nit za unos znakova sa tastature, treću nit za proveru pravopisne i gramatičke ispravnosti unetog teksta.

Korišćenje višenitnog koncepta ima sledeće prednosti:

- **Smanjanje vremena odziva** – višenitna tehnika omogućava interaktivnim aplikacijama da nastave rad, čak i kada je deo programa blokiran ili izvršava neku dugotrajnu operaciju. Npr., višenitni Web prikazivač može da nastavi interakciju s korisnikom na jednoj niti, dok druga nit simultano učitava neki sliku sa Interneta
- **Ekonomičnost** – ogleda se u deljenju resursa i prostora i uštedi vremena koje takođe utiče na performanse. Niti dele memoriju i sve ostale resurse koji pripadaju istom procesu. Niti se pravrnog brže od procesa, a prebacivanje konteksta između niti takođe je brže od prebacivanja konteksta između prcesa.

- **Bolje iskorišćenje višeprocesorske arhitekture** – bilo koje niti mogu se istovremeno izvršavati, svaka na različitom procesoru. Ukoliko imamo samo jedan proces u sistemu, jednonitna koncepcija dozvoljava izvršavanje tog procesa samo na jednom procesoru, dok su ostali procesori neiskorišćeni.

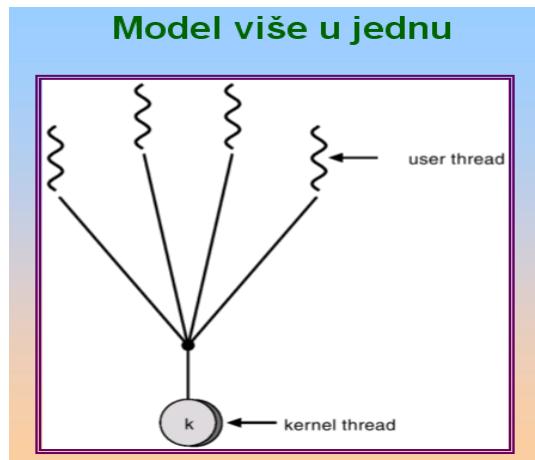
Korisničke niti i niti jezgra

Kako jedan proces može da se izvršava i u korisničkom režimu i u režimu jezgra, potrebno je obezbediti istu podršku i na korisničkom nivou za korisničke niti (*user threads*) i na nivou jezgra za niti jezgra (*kernel threads*). Podrška za korisničke niti se realizuje preko biblioteke za rad sa korisničkim nitima (*user thread library*). Ova biblioteka obezbeđuje podršku za stvaranje niti, raspored izvršavanja niti i upravljanje nitima, ali bez uticaja jezgra. Najznačajnije vrste korisničkih niti su: POSIX Pthreads, Mach C-threads, Solaris UI-threads.

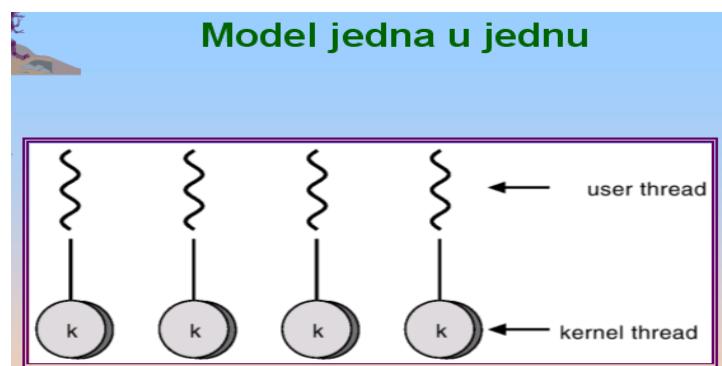
Niti jezgra direktno podržava operativni sistem. Jezgro izvršava operacije stvaranja niti, raspoređivanja izvršavanja niti i upravljanja nitima u prostoru jezgra. Niti jezgra se prave sporije, a jezgro ozbiljnije i efikasnije upravlja svojim nitima nego bibliotekama za rad s korisničkim nitima.

Mnogi sistemi podržavaju obe vrste niti, a zavisno od toga kako se korisničke niti mapiraju (preslikavaju) u niti jezgra, postoje tri glavne koncepcije, tj tri višenitna modela:

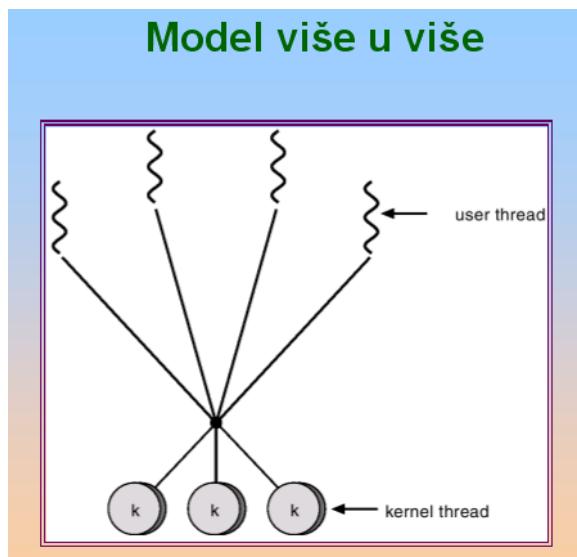
- **Model više u jedan (many-to-one)** - U ovom modelu, više korisničkih niti mapira se u jednu nit jezgra. Upravljanje nitima se odvija na korisničkom nivou i može biti efikasno, ali ima nedostatke. Ako neka od niti obavi bilo koji blokirajući sistemski poziv, blokiraće se ceo proces, to jest sve njegove niti. S obzirom na to da samo jedna nit može pristupiti jezgru u jednom trenutku, u režimu jezgra neće moći da se iskoristi višeprocesorska arhitektura. Ovaj model koristi operativni sistem Solaris i biblioteku Green threads.



- **Model jedna u jednu (one-to-one)** - U ovom modelu, koji je karakterističan za operativne sisteme Windows NT, Windows 2000/XP/2003 i OS/2, svaka korisnička nit mapira se u jednu nit jezgra. Ovaj model obezbeđuje mnogo bolje konkurentno izvršavanje niti, dozvoljavajući da druge niti nastave aktivnosti u slučaju kada jedna nit obavi blokirajući sistemski poziv. Takođe se omogućava da se više niti jezgra izvršavaju paralelno na višeprocesorskoj arhitekturi. Kako se mnogo vremena gubi pri stvaranju i održavanju niti jezgra, mnogi operativni sistemi ograničavaju maksimalan broj niti jezgra.



- **Model više u više (many-to-many)** - U ovom modelu, više korisničkih niti mapira se u manji ili isti broj niti jezgra, pri čemu mapiranje zavisi od operativnog sistema a naročito od broja procesora. Ovo je najkompleksniji i najkvalitetniji model jer može da se prilagodi i aplikaciji i procesorskom okruženju. Ovaj više nitni model zastupljen je u sistemima Solaris-2, IRIX, HP-UX i True64 UNIX.



Rad sa nitima

U više nitnoj arhitekturi menja se semantika sistemskih poziva za upravljanje procesa i rad s njima. Mora postojati poziv za pravljanje teškog procesa, koji će duplicirati sve niti i njihove adresne prostore, kao i poziv za pravljenje jedne jedine niti.

Prekidanje niti je aktivnost koja prekida izvršenje niti pre njenog prirodnog završetka. Nit koja će biti prekinuta obično se naziva ciljna nit (*engl. target thread*). Prekidanje ciljne niti može se dogoditi po dva scenarija:

- **asinhroni prekid (engl. asynchronous cancellation)** - aktivnost niti se odmah prekida bez obzira na stanje u kome se nalazi;
- **odloženi prekid (engl. deferred cancellation)** - ciljna nit periodično proverava treba li da prekine aktivnost. Ako treba, to će učiniti u povoljnem trenutku – kad završi neku celinu, tj. obavi deo posla do kraja.

Odloženo prekidanje je novije i kvalitetnije, ali se teže implementira, tako da većina operativnih sistema primenjuje asinhrono prekidanje.

Signal upućen nekom višenitnom procesu, zavisno od situacije, može se proslediti:

- samo onoj niti kojoj je namenjen
- svim nitima procesa
- samo nekim nitima procesa
- samo onoj niti procesa čija je namena da prima i obrađuje sve signale za proces