

Sistemske pozivi i programski interfejs *Unix*-olikih operativnih sistema

Operativni sistemi

Fakultet tehničkih nauka, Univerzitet u Novom Sadu

18.01.2021.

Sistemi pozivi

- Sistemi pozivi su interfejs preko kojeg programi mogu da komuniciraju sa jezgrom (*kernel*-om) operativnog sistema.
- *Unix*-oliki operativni sistemi su ogromnom većinom implementirani u programskom jeziku C, te su i njihovi sistemski pozivi predviđeni da se koriste u tom programskom jeziku.

Sistemiški pozivi

- Postoji više kategorija osnovnih sistemskih poziva:
 - Sistemiški pozivi za rad sa datotekama.
 - Sistemiški pozivi za rad sa procesima.
 - Sistemiški pozivi za rad sa memorijom.
 - Sistemiški pozivi za međuprocesnu komunikaciju.

Rad sa datotekama

- Sistemski pozivi za rad sa datotekama:
 - *stat* – Vraća informacije o datoteci.
 - *open* – Otvara datoteku i vraća njen deskriptor.
 - *creat* – Kreira i otvara datoteku i vraća njen deskriptor.
 - *read* – Čita podatke preko deskriptora.

Rad sa datotekama

- Sistemski pozivi za rad sa datotekama:
 - *write* – Upisuje podatke preko deskriptora.
 - *lseek* – Pomera trenutnu poziciju za čitanje/pisanje u deskriptoru.
 - *dup* – Duplicira deskriptor.
 - *select* – Čeka dok deskriptor ne postane dostupan za neku operaciju (čitanje ili pisanje).

Rad sa procesima

- Sistemski pozivi za rad sa procesima:
 - *fork* – Klonira pozivajući proces.
 - *exec* – Postavlja novi program u pozivajući proces.
 - *wait* – Čeka da kreirani proces završi sa radom.
 - *exit* – Završetak rada pozivajućeg procesa.
 - *getpid* – Vraća PID pozivajućeg procesa.
 - *getppid* – Vraća PID procesa koji je kreirao pozivajući proces.

Fork

- Sistemski poziv *fork* kreira novi proces od pozivajućeg procesa.
- Novi proces je identičan originalnom procesu u trenutku povratka iz sistemskog poziva, sa izuzetkom povratne vrednosti.
 - Originalni proces dobija PID novog procesa kao povratnu vrednost.
 - Novi proces dobija 0 kao povratnu vrednost.

- Memorija novog procesa je identična memoriji originala, ali se naknadne promene ne sinhronizuju.
- Operativni sistemi uglavnom ne prave duplikat celokupne memorije procesa, već zbog efikasnosti koriste tehniku *copy-on-write*.

Rad sa procesima

- Sistemski poziv `execve` izvršava program u procesu koji poziva.
- Programu se mogu proslediti argumenti komandne linije i promenljive okruženja.
- Celokupna memorija procesa koji pozove ovu funkciju bude zamenjena sa novim programom.

Rad sa procesima

- Sistemski poziv *wait* prebacuje pozivajući proces u stanje čekanja sve dok se ne promeni stanje nekog od procesa koje je pozivajući proces pokrenuo.
- Kada proces završi sa radom, informacija o njemu ostaje u memoriji dok proces koji mu je “roditelj” ne pozove ovaj sistemski poziv.
- Ugašen proces kojem još nije pročitana povratna vrednost se zove i “*zombi*” proces.

Rad sa memorijom

- *brk* – Pomeri granicu memorije pozivajućeg procesa.
- *mmap* – Mapira sadržaj datoteke u memoriju pozivajućeg procesa.

Međuprocesna komunikacija

- *signal* – omogućava procesima da postavе svoje obrađivače signala, za bilo koji signal osim SIGSTOP i SIGKILL.
- *kill* – šalje signal datom procesu.
- *pipe* – otvara *FIFO* kanal za komunikaciju sa dva kraja, ulaz sa jednog kraja se pojavljuje kao izlaz na drugom kraju, i suprotno.