

Unix

Večuporabniški

Večopravilni

Osnovne funkcije

- zagotoviti interface med rač. in upor.
- upravljati IO operacije
- nalaganje in izvrševanje programov
- upravljanje datot. sistema
- upravljanje s time-sharingom in večopr.
- upravljanje in dodeljevanje pomnilnika

Time sharring

- v izbranem čas. trenutku v sistemu več procesov
- procesorjev čas med te procese porazdeljen

Pridobitve

- več uporabnikov hkrati : zmanjša stroške procesiranja
- več opravil naenkrat : večja produktivnost

Zgradba sistema (groba)

- večuporabniški OS
- večje število programskih orodij
- precejšnje število aplikativnih orodij

Jedro OS

- opravlja vso aktivnost oz. funkcionalnost
- majhen : rezidenčen v GP
- z okoljem komunicira preko sistemskih klicev

Sistemiški klici

- upravljanje z datotečnim sistemom
- nadzor programskih procesov
- medsebojna komunikacija procesov

Pojmi

- Programski proces
- Slika programskega procesa
- Izvajanje procesa
- Kreiranje procesa

Programski proces

- izvajanje zaporedja instrukcij na dani množici podatkov
- življenska doba procesa

Življenska doba procesa

- proces nekdo tvori in uničuje
- stanje izvajanja
- stanje čakanja na izvajanje
- vsaj en proces kreiran ob zagonu sistema
- sistemski proces
- uporabniški proces

nadaljevanje...

- v vsakem trenutku v izvajanju le en proces ostali v stanju čakanja

Slika procesa

- celota podatkov o procesu, ki se v danem trenutku shrani
- sestavljajo jo :
 - ◆ vsebina pomnilnika(podatki+instr.)
 - ◆ vsebina strojnih registr.
 - ◆ stanje odprtih datotek
 - ◆ accounting

Ugotovitev

- programski proces je izvajanje slike programskega procesa
- slika se lahko nahaja v GP
- ali pa v sekundarnem pomnilniku
- ali pa v sek+GP

Izvajanje procesa

- med izvajanjem je proces v pomnilniku predstavljen s tremi ločenimi sekcijami :
 - ◆ tekst (koda), WP, sh.
 - ◆ data (podatki)
 - ◆ stack (sklad)
- slednji sta lastni procesu

Nastanek procesa

- aktivni proces izvede sistemski klic `fork()`
- ob rojstvu je otrok kopija starševskega procesa, loči se le po identifikacijski številki PID
- proces oče pozna pid otroka in tako lahko nanj vpliva
- `kill()`, `exit()`