

I/O System

MMM Što je I/O System

I/O System je naziv za ustav koji omogućava prihvata i emitiranje signala ili podataka.

I/O System se sastoji od sklopovlja i programske podrške

MMM Čemu služi programska podrška za I/O System

Programska podrška za I/O System odvaja aplikacijsku programsku podršku od sklopovlja.

Programska podrška za I/O System ostvaruje razinu apstrakcije u ulazno-izlazne operacije aplikacijske programske podrške.

MMM Koji su zahtjevi na programsku podršku za I/O System

Programska podrška za I/O System treba biti modularna, slojevita, skalabilna i uniformna.

SW za I/O System je modularan ako se sav programski kod za upravljanje jednom vrstom međusklopa nalazi isključivo u jednom programskom modulu (ili skupini).

SW za I/O System je slojevit ako se upravljanje međusklopom nalazi u sloju programskih modula koji je odvojen od onoga koji se bavi reprezentacijom podataka, te onoga koji se bavi oblikovanjem i interpretacijom podataka, a između tih slojeva je jasno definiran način povezivanja i protoka podataka.

SW za I/O System je skalabilan ako je za upravljanje s više istovrsnih međusklopova dovoljna samo jedna instanca odgovarajućeg programskog modula.

SW za I/O System je uniforman ako se svim vrstama ulazno-izlaznih međusklopova upravlja na jednak način.

MMM Apstrakcije i slojevi

SW slojevi za I/O System su aplikacija, korisnička biblioteka I/O modula, jezgri pozivi (system calls), device driveri i prekidne rutine.

Korisnička biblioteka I/O modula prevodi specifične potrebe i vrste I/O poslova na jednoobrazne kernel pozive.

Korisničke biblioteke I/O modula su dio aplikacijskog okruženja i proizvode ih proizvođači programskih alata, dok su jezgri pozivi i device driveri dio operacijskog sustava.

Jezgreni pozivi (kernel calls / system calls) su isti za sve tipove međusklopova.

Jezgreni pozivi (kernel calls / system calls) potpuno odvajaju korisnika/aplikaciju od sklopovlja.

Device driver se gradi posebni za svaku vrstu međusklopa.

Jedan device driver upravlja s više istovjetnih međusklopova/uređaja.

Device driver se ne bavi ulazom/izlazom pojedinačnih podataka već samo priprema posao interrupt rutinama, koje se time bave.

U najužem smislu, samo su jezgreni pozivi dio operacijskog sustava, jer device driver u načelu proizvodi proizvođač sklopovlja.

MMM ***I/O moduli kernela***

I/O moduli kernela sadrže pet osnovnih rutina: open, close, read, write i ioctl.

Pet osnovnih rutina I/O modula kernela su istovjetne za sve fizičke uređaje i odgovarajuće međusklopove.

Sve složenije I/O funkcije moraju se kombinirati iz pet osnovnih rutina I/O modula kernela (ali na razini korisničkih biblioteka).

I/O moduli kernela zapravo ne izvršavaju svoje funkcije, već njihovo izvršenej prenose na korespondentne funkcije device drivera. Njihov je zadatak da indentificiraju i pozovu pravi device driver za određenu vrstu međusklopa te da mu rpenesu podatak o kojoj instanci istovrsnog međusklopa se radi. Na primjer, treći disk.

Open rutina služi za pripremanje sklopovlja za rad. Na primjer, uključuje motor floppy diska.

Open rutina ponekad može i alocirati potrebnu memoriju za rad.

Open rutina se poziva sa simboličkim imenom uređaja.

Prilikom poziva Open rutine definira se je li otvaranje uređaja za čitanje, pisanje ili oboje.

Prilikom poziva Open rutine definira se je li otvaranje uređaja s blokiranjem ili bez, kada nema ulaznih podataka ili se ne može izvršiti pisanje.

Ako se prilikom poziva Open rutine definira otvaranje uređaja s blokiranjem, to znači da se kod čitanja/pisanja kontrola neće vratiti procesu koji je pozvao Open rutinu sve dok se ne obavi čitanje/pisanje željenog broja podataka.

Ako se prilikom poziva Open rutine definira otvaranje uređaja bez blokiranja, to znači da će se kod čitanja/pisanja kontrola vratiti procesu koji je pozvao Open rutinu zajedno s podatkom o broju pročitanih/zapisanih podataka, koja može biti i nula.

Close rutina služi zaustavljanju sklopovlja. Na primjer, može ugasiti motor floppy diska.

Ako je Open alocirao memoriju, Close ju obvezno mora osloboditi.

Close se poziva sa simboličkom oznakom otvorenog kanala, dobivenom prilikom Open poziva.

Close oslobađa, za vrijeme upotrebe postavljena, zauzeća

Read poziv učitava željeni broj bajtova s uređaja.

Read se poziva sa simboličkom oznakom otvorenog kanala, brojem podataka koje treba učitati i memorijskom adresom na koju treba spremi učitane podatke.

Write poziv zapisuje željeni broj bajtova na uređaja.

Write se poziva sa simboličkom oznakom otvorenog kanala, brojem podataka koje treba zapisati i memorijskom adresom na kojoj se nalaze podaci koje treba zapisati.

ioctl poziv obavlja upravljanje međusklopom.

Upravljanje pomoću ioctl poziva je specifično za svaku pojedinu vrstu međusklopa.

ioctl poziv zapravo samo prosljeđuje naredbe i strukture device driveru

MMM **Device Drivers**

Device driver se gradi posebno za svaku pojedinu vrstu međusklopa.

SW rutine device drivera su korespondentne system callovima: open, close, read, write i ioctl.

Read i Write rutine device drivera zapravo ne čitaju i ne pišu, već pripremaju posao za prekidnu rutinu koja upravlja sklopovljem koje čita ili piše.

Read i Write rutine device drivera pokreću postupak čitanja ili pisanja, a onda završetak prvog podzadatka (npr. prijem jednog znaka) pokreće prekidnu (IRQ) rutinu koja pokreće slijedeći podzadatak i tako sve do posljednjeg znaka.

Read i Write rutine device drivera su zapravo te koje rade u blocking ili nonblocking režimu rada.

MMM **Problem nedjeljivog resursa**

Problem nedjeljivog resursa se odnosi na neke uređaje poput: printer, zvučnik, disk.

Problem nedjeljivog resursa se javlja kad bi istovremeno, isprepletano korištenje istog uređaja od strane više procesa: izmiješalo podatke, mijenjalo stanje uređaja/resursa i vršene funkcije ili mijenjalo podešenost, konfiguraciju uređaja.

Kod problema nedjeljivog resursa se javlja potreba za prioritetima.

Kod problema nedjeljivog resursa se javlja potreba za nonblocking radom.

Spooling znači "Simultaneous Peripheral Operations On-line".

Kod Spoolinga samo jedan, posebni korisnički program (spooler) pristupa periferiji.

Kod Spoolinga svi korisnici (programi) se suzdržavaju od izravne upotrebe perifernog uređaja i umjesto toga ostavljaju svoje zahtjeve na djeljivom mediju.

Spooler raspodjeljuje zahtjeve na resurs (periferni uređaj) prema unaprijed dogovorenim pravilima