Uvod u informatiku
- Ulazno – Izlazni uređaji -

Veljko Stanković

Uvod

- Dva najvažnija zadatka računarskih sistema jesu obrada podataka i upravljanje ulazno-izlaznim (I/O) uređajima.
- OS ima zadatak da upravlja i kontroliše rad I/O uređaja.
- Upravljanje I/O operacijama predstavlja jedan od glavnih zadataka OSa pošto se funkcionalnost i brzina različitih I/O uređaja mogu veoma razlikovati i varirati.
- Možemo identifikovati dva kontradiktorna trenda u razvoju I/O uređaja.
  - Kao prvo, javlja se potreba da se standardizuje interfejs preko koga se povezuju I/O uređaji.
  - Na drugoj strani, javlja se problem kako inkorporirati nove uređaje koji se mogu veoma razlikovati od ranije razvijenih uređaja za I/O.

Uvod

- Računarski sistem upravlja velikim brojem različitih I/O uređaja.
- Najčešće se koriste
  - uređaji za skladištenje podataka (diskovi, trake),
  - komunikacioni uređaji (mrežne kartice, modemi) i
  - uređaji za komunikaciju sa čovekom (displej, tastatura, miš).
- I/O uređaj komunicira sa računarskim sistemom tako što razmenjuje podatke preko vodova ili bežičnim putem.
  - Tačka u kojoj se I/O uređaj vezuje sa računarskim se zove port.
  - Ako dva i više uređaja dele jedan skup vodova, tada se takva se takva veza zove magistrala.
  - Blok koji kontroliše rad porta, magistrale ili uređaja se zove kontroler.
    - ☑ Neki uređaji imaju na sebi ugrađene kotrolere kao na primer HDD.
  - Svaki kontroler ima registre preko kojih razmenjuje podatke i kontrolne poruke sa CPU.

Uvod

- Prema načinu adresiranja I/O uređaja razlikujemo:
  - Preslikani I/O prostor
    - ☑ Kada glavna memorija i I/O uređaji dele adresni prostor CPU.
    - ☑ U ovom slučaju se za komunikaciju sa I/O uređajima koriste standardne naredbe za upis i čitanje podataka.
  - Izdvojeni I/O prostor
    - ☑ Kada su adresni prostori glavne memorije i I/O uređaja izdvojeni.
    - ☑ Naredbe za upis i čitanje podataka u glavnu memoriju i naredbe za komunikaciju sa I/O uređajima se razlikuju.
- Moguće je kombinovati ova dva pristupa.

Uvod

- I/O port se obično sastoji iz sledećih registara:
  - statusni,
  - kontrolni, ulazni i
  - izlazni registar.
- Statusni registar omogućava da CPU utvrdi da li su podaci dostupni za čitanje ili je došlo do grške.
- U kontrolni registar CPU upisuje komande ili menja stanje I/O uređaja.

Upravljanje I/O sistemom

1010011
1110100
1100001
1010110

— Prema načinu kontrole I/O uređaja razlikujemo:

- Programirani I/O
  - ☑ Prenos podataka između glavne memorije i I/O uređaja kontroliše CPU.
  - ☑ CPU mora periodično prema određenom prioritetu da prozove (pooling) svaki od I/O uređaja i da vidi da li imaju neke podatke ili se promenio njihov status.
- Prekidački I/O
  - ☑ Kada neki I/O uređaj ima podatke koje bi terbalo da preda CPU ili se njegov status promeni, I/O generiče određeni prekidački signal kojim signalizira CPU i inicira obradu od strane CPU.
  - ☑ CPU više ne troši dragoceno vreme na prozivanje pojedinačnih I/O uređaja kako bi proverio njihovo stanje, ali i dalje aktivno učestvuje i kontroliše komunikaciju sa njima.
  - ☑ Interapti mogu biti višeg iil nižeg prioriteta, maskirajući ili nemaskirajući.

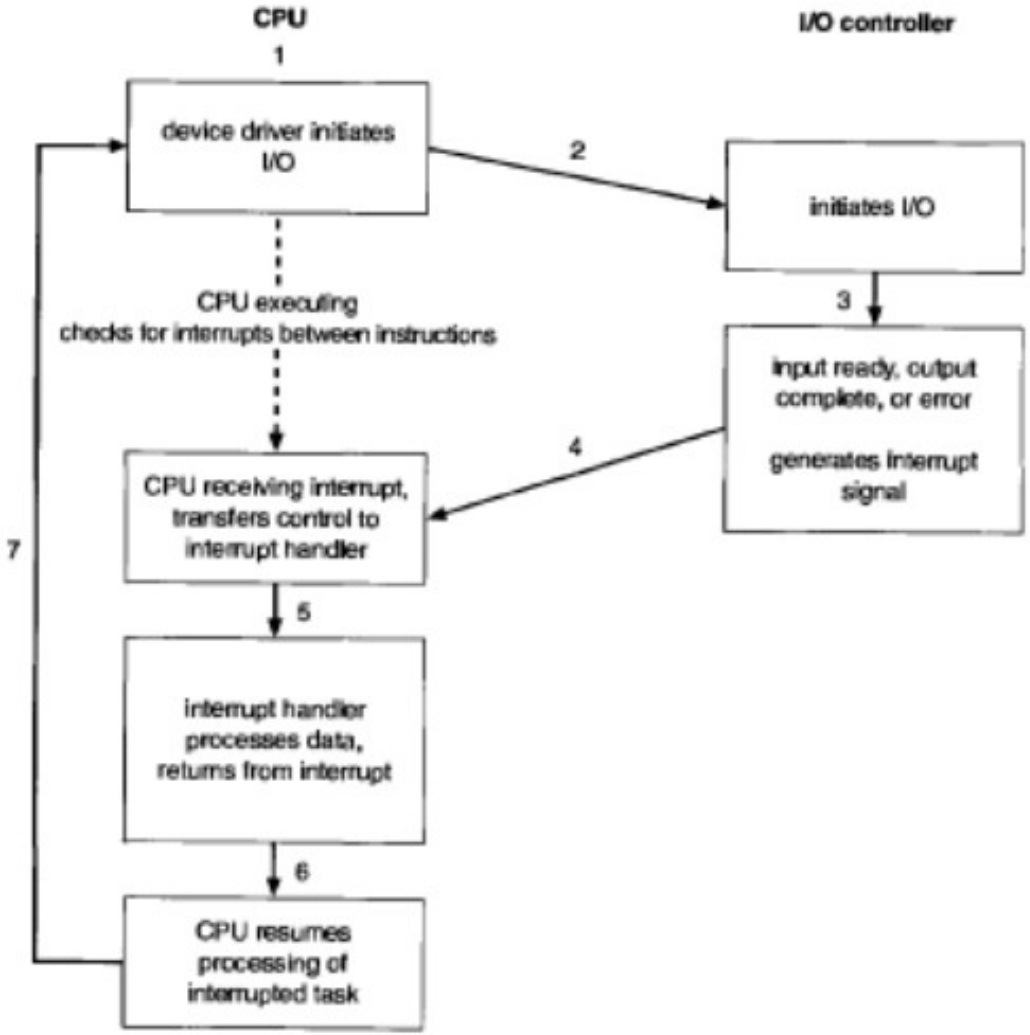
Upravljanje I/O sistemom

1010011
1110100
1100001
1010110

- DMA (Direct Memory Access)
  - Kada se javi potreba za komunikaciju I/O uređaja i računarskog sistema, I/O uređaj aktivira prekidački signal (interapt) kojim signalizira CPU.
  - CPU proverava stanje I/O uređaja i potom predaje kotrolu nad njim posebnom bloku koji se zove DMA kontroler.
  - Nakon što se završi prenos podataka, DMA kontroler šalje interapt CPU.

Mehanizam obrade prekida

1010011
1110100
1100001
1010110



Komunikacija aplikacija i I/O uređaja

1010011
1110100
1100001
1010110

- Posmatramo tehnike i interfejse ka OS koji omogućavaju jednoobrazan tretman I/O uređaja, tj. kako aplikacije komuniciraju sa I/O uređajima nezavisno od njihove prirode i načina funkcionisanja.
  - To se postiže uz pomoć:
    - ☑ abstrakcije,
    - ☑ enkapsulacije i
    - ☑ hijerarhijske organizacije softvera.
- Drajveri I/O uređaja predstavljaju softver koji imaju ulogu da sakriju razlike pojedinih kontrolera I/O uređaja od I/O podsistema kernela.
- Time što smo učinili I/O podsistem nezavisnim od hardvera, značajno smo pojednostavili proces razvoja OSa.

Komunikacija aplikacija i I/O uređaja

1010011
1110100
1100001
1010110

— I/O uređaji se razlikuju po:

- Prenos podataka se vrši u blokovima ili kao sekvenca karaktera.
- Pristup podacima je sekvencijalan ili direktan. Pri sekvencijalnom prenosu podataka do I/O uređaja, podaci se prenose po određenom redosledu.
- Kod I/O uređaja sa direktnim pristupom, korisnik može da pristupi bilo kom podatku direktno.
- Sinhrona ili asinhrona komunikacija.
- Deljiv ili nedeljiv I/O uređaj.
  - ☑ Deljivom I/O uredjaju mogu da pristupe više procesa, dok se nedeljivi uređaj može dodeliti samo jednom procesu.
- Brzina rada I/O uređaja.
- Na I/O uređaju se mogu podaci samo upisivati, čitati ili čitati i upisivati.

Komunikacija aplikacija i I/O uređaja

1010011
1110100
1100001
1010110

- Iako se sistemski pozivi mogu razlikovati među operativnim sistemima, kategorije I/O uređaja su prilično standardizovane.
- Konvencinalni pristup organizaciji I/O uključuje:
  - blok I/O,
  - sekvencijalni I/O,
  - memorijski preslikani pristup fajlovima i
  - mrežni socket.

Blok i sekvencijalni I/O uređaji

1010011
1110100
1100001
1010110

- Interfejs ka blok uređajima obuhvata sve aspekte neophodne za pristup diskovima i drugim blok I/O uređajima.
- Predpostavlja se da uređaj može da primi komande read, write i seek u slučaju kada ima mogućnost direktnog pristupa podacima.
- Memorijski preslikani pristup podacima može biti implementiran na osnovu drajvera za blok orijentisane I/O uređaje.
  - Umesto read i write operacija, memorijski preslikani interfejs omogućava pristup podacima na disku preko lokacija u glavnoj memoriji.
  - Pošto se prenos podataka vrši uz pomoć istih mehanizama koji se koriste kod virtuelne memorije, memeorijski prelsikani pristup I/O uređajima je efikasan.
  - Memorijski preslikani pristup I/O uređajima je pogodan i za programere jer je pristup fajlu jednostavan poput upisa i čitanja iz memorije.
  - OS uobičajeno nude virtuelnu memoriju kako bi se omogućila primena memorijski preslikanog interfejsa.

Blok i sekvencijalni I/O uređaji

- Tastatura predstavlja primer sekvencijalnog I/O uređaja.
  - Osnovni sistemski pozivi get i set omogućavaju upis i čitanje podataka kod ovakvih interfejsa.
  - Ovakav pristup je pogodan za uređaje kao što su tastatura, miš ili modem.

Mrežni uređaji

- Socket interfejs za komunikaciju se koristi kod velikog broja operativnih sistema uključujući UNIX i Windows.
- Sistemski pozivi omogućavaju aplikaciji da kreira socket, da poveže lokalni socket sa tačkom sa određenom adresom i da prima i šalje poruke ka nekoj aplikaciji preko tog socketa.
- Sistemski poziv select kojom upravljamo grupom socketa.
  - Ovaj poziv omogućava da utvrdimo koji od socketa imaju pakette koji očekuju prijem, čime eliminišemo potrebu za prozivanjem socketa ili uposleno čekanje.

Blokirajući i neblokirajući I/O

- Blokirajući i neblokirajući I/O
  - Asinhroni i sinhroni I/O
- Blokirajući I/O
  - U slučaju blokirajuće I/O operacije, kada aplikacija aktivira blokirajući I/O, njeno izvršavanje se prekida i ona se smešta u waiting red čekanja.
  - Nakon izvršavanja I/O operacije aplikacija nastavlja sa izvršavanjem.
  - Aktivnosti I/O uređaja su uopšte asinhrone, tj. vreme njihovog izvršavanja je nepredvidivo.
  - Većina OSa i pored toga koriste blokirajuće I/O pozive jer su lakši za razumevanje od neblokirajućih.
- Jedan od načina da se obezbedi istovremeno izvršavanje na CPU i izvršavanje I/O operacija jeste primenom niti.

Blokirajući i neblokirajući I/O

- Drugi način komunikacije sa periferijom jeste primenom neblokirajućih (asinhronih) I/O poziva.
- Neblokirajući I/O poziv ne prekida izvršavanje aplikacije već nastavlja sa izvršavanjem pri čemu vraća informaciju koji deo podataka je uspešno prenesen.
- Asinhroni I/O poziv ne prekida izvršavanje aplikacije, a po završetku I/O operacije rezultat se šalje aplikaciji koristeći mehanizam trapa ili promenom vrednosti neke promenljive u adresnom prostoru aplikacije.
- Razlika između blokirajućeg I/O poziva i asinhronog I/O poziva jeste u tome što asinhroni poziv odmah vraća upravljanje aplikaciji sa raspoloživim podacima, dok sinhtoni I/O poziv zahteva prenos podataka koji će biti u potpunosti izvršen.

Raspoređivanje I/O operacija

- Jedan od zadataka I/O podsistema jeste da nađe najbolji redosled izvršavanja I/O operacija.
- Raspoređivanje I/O operacija može poboljšati ukupne performanse sistema, omogućući ravnopravan pristup I/O uređajima i smanji srednje vreme čekanja na izvršenje I/O operacija.
- Ova funkcija se implementira tako što se za svaki I/O uređaj formira red čekanja.
- Kada aplikacija aktivira blokirajući I/O poziv zahtev se smešta u red čekanja za taj uređaj.
- I/O alokator vrši preraspodelu procesa u redu čekanja za neki I/O uređaj kako bi poboljšali efikasnot iskorišćenja sistema.
- Takođe, pri preraspodeljivanju treba težiti da pristup I/O uređajima bude što ravnopravniji.

Strukture podataka kernela

1010011
1110100
1100001
1010110

- Kernel mora da čuva informaciju o stanju upotrebe I/O komponenti.
- To čini primenom niza kernel struktura podataka kao što je tabela otvorenih fajlova.
- Kernel koristi dosta sličnih struktura kako bi pratio stanje komunikacionih veza, sekvencijalnih I/O uređaja i drugih I/O aktivnosti.
- Fajl sistem UNIX sistema ima mogućnost pristupa nizu entiteta kao što su korisnički fajlovi, periferni uređaji i adresnom prostoru procesa.
  - Svi softverski entiteti se posmatraju kao fajlovi.
  - Tabela otvorenih fajlova sadrđi pokaznu tabelu u kojoj se čuvaju pointeri ka odgovarajućim rutinama u zavisnoti od tipa fajla.

Strukture podataka kernela

— I/O podsistem kernela nadgleda:

- Upravlja domenima imena fajlova i uređaja.
- Kontroliše pristup fajlovima i uređajima.
- Kontroliše zahteve za obradom.
- Dodela prostora fajlovima.
- Dodela I/O uređaja.
- Baferovanje, upravljanje keš meorijom i spulovanje.
- Raposređivanje I/O zahteva.
- Nadlgedanje stanja I/O uređaja, kontrola grešaka i opravak od otkaza.
- Konfiguracija i primena drajvera I/O uređaja.

Transformacija I/O zahteva u hardverske operacije

1010011
1110100
1100001
1010110

- Tokom podizanja operativnog sistema, tzv. butovanje (boot), sistema testira koje su komponente računarskog sitema prisutne i potom učitava potebne drajvere uređaja, bilo na početku rada sistema ili kada se javi prvi I/O zahtev ka određenom I/O uređaju.
- Posmatramo primer pristupa podacima na disku.
  - Kod MS DOS/Windows, podacima se pristupa tako što se na osnovu putanje do fajla određuje particija i direktorijum a potom se u FAT tabeli određuje adresa blokova na HDD.
  - Nasuprot tome, kod UNIX-a i UNIX klonova, svi fajlovi i uređaji se posmatraju na jedinstveni način.
    - ☑ Na osnovu stabla direktorijuma i mount tabele se određuje inode, tj. indeks blok, koji sadrži adrese blokova na HDD koji su dodeljeni datom fajlu.
    - ☑ Ako se radi o I/O uređaju, tada struktura direktorijuma umesto inode sadrži zapis u formatu <major, minor>, gde major služi za identifikaciju drajevra koji trebaju da opsluže dati I/O zahtev, dok minor predstavlja broj I/O uređaja na osnovu kojeg se u tabeli I/O uređaja određuje adresa porta uređaja ili memorijski preslikanog prostora.

Transformacija I/O zahteva u hardverske operacije

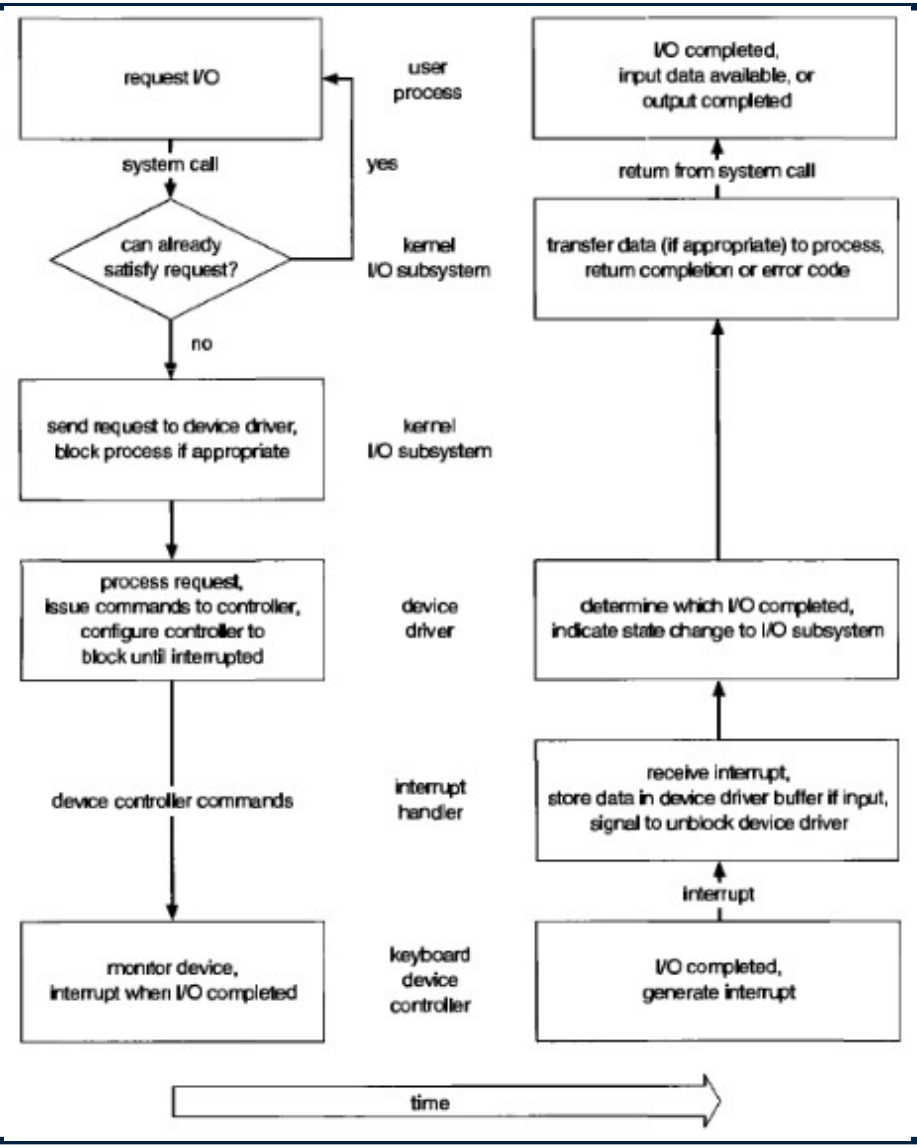
- Tipičan ciklus obrade blokirajućeg zahteva za učitavanje podataka sesastoji iz sledećih faza:
  - Proces zahteva blokirajući read sistemski poziv za fajl koji je prethodno otvoren pozivom open.
  - Kod za obradu sistemskih poziva u kernelu proverava korektnost parametara. U slučaju upisa odataka, proverava se keš memorija, podaci se predaju procesu i završava se I/O poziv.
  - Ako se podatak ne nalazi u keš memoriji pristupa se sekundarnoj memoriji. Proces se premešta iz ready u red čekanja za dati I/O red čekanja i vrši se alokacija I/O zahteva. Posle odeđenog vremena, I/O podsistem šalje zahtev odgovarajućem drajveru uređaja. U zavisnosti od OS-a, zahtev se šalje u vidu kernel poruke ili se izvršava u vidu subrutine.
  - Drajver uređaja alocira deo memorije za bafer preko koga prima podatke i raspoređuje I/O aktivnost. Posle određenog vremena, drajver šalje komande kontroleru uređaja tako što ih upisuje u kotrolne registre uređaja.

Transformacija I/O zahteva u hardverske operacije

- Tipičan ciklus obrade blokirajućeg zahteva za učitavanje podataka sesastoji iz sledećih faza:
  - Kontroler aktivira hardver I/O uređaja kako bi izvršio transfer podataka.
  - Drajver može da očita stanje i podatke sa uređaja ili može da inicira DMA prenos podataka u memoriju. Predpostavićemo da prenosom podataka upravlja DMA kontroler koji generiše prekid kad se proces prenosa završi.
  - Aktivira se odgovarajuća rutina za obradu prekida na osnovu vektora prekida, skladište se potrebni podaci i signalizira odgovarajućem drajveru.
  - Drajver uređaja prima signal, utvrđuje koji I/O zahtev je opslužen, određuje status zahteva i signalizira I/O podsistemu kernela da je zahtev opslužen.
  - Kernel prenosi podatke ili vraća kod u adresni prostor procesa koji je aktivirao zahtev i prebacuje taj proces iz wait reda čekanja u ready red čekanja.
  - Prebacivanjem procesa u ready red čekanja deblokiramo taj proces. Kada alokator dodeli CPU tom procesu, on nastavlja izvršavanje od tačke gde je aktivirao sistemski poziv.

Transformacija I/O zahteva u hardverske operacije

1010011
1110100
1100001
1010110



Domaci zadatak

1010011
1110100
1100001
1010110