

Željko Milović

Miloš Doroslovački

Aleksandar Zavaljevski

INSTITUT "BORIS KIDRIČ" - VINČA

OOOR Institut za računarsku tehniku "RT"

P. fah 522, 11001 Beograd

UPRAVLJANJE UREDJAJIMA MASOVNE MEMORIJE U
SISTEMU SA VIŠE MIKROPROCESORA

DEVICE MANAGEMENT OF MASS STORAGE DEVICES
IN A MULTIMICROPROCESSOR SYSTEM

SADRŽAJ - U radu je prikazan ulazno/izlazni sistem jednog operativnog sistema za potrebe složenih mikror računarskih sistema. Ukratko je opisana arhitektura hardvera na kojoj se izvršava U/I sistem. Data je detaljna analiza obrade U/I zahteva u sistemu, sa posebnim osvrtom na iskorišćenje vremena čekanja jednog procesa na obradu jednog U/I zahteva za izvršavanje drugih procesa. Opisani su moduli U/I sistema sa naglaskom na sinhronizacione mehanizme i korišćenje primitiva nukleusa operativnog sistema. Dat je prikaz osnovnog razvojnog puta od upravljačkih programa do jednog zatvorenog U/I sistema. Opisan je jednostavan interfejs ka višim slojevima operativnog sistema i aplikacionim procesima.

ABSTRACT - The paper deals with Input/Output System of an operating system designated for complex microcomputing systems. The main features of hardware architecture where I/O system is running are described. A detailed analysis of an I/O request servicing is given, especially with regard to the I/O wait time. Modules of I/O system are described, with emphasize on the synchronization mechanisms and the use of operating system nucleus primitives. The basic development cycle from the device drivers to a complete I/O system is presented. A simple interface to the both upper levels of operating system and the application tasks is described.

1. UVOD

U ovom radu je opisan problem izrade ulazno-izlaznog sistema za upravljanje uredjajima masovne memorije u sistemu sa više specijalizovanih mikroprocesora. U/I sistem treba da bude deo multiprogramskog operativnog sistema. Teorijski rezultati iz oblasti ope-

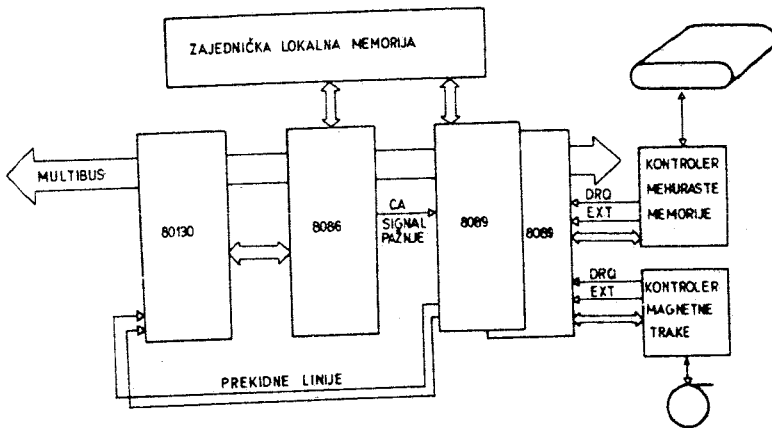
rativnih sistema na koje se ovaj rad oslanja pre svega se odnose na problematiku paralelizacije izvršavanja relativno apstraktnih zahteva, konkurentnog programiranja i testiranja u procesu pisanja programskog sistema koji je inherentno sastavljen iz više procesa. Praktični rezultati na koje se rad oslanja se pre svega odnose na modele sistema sa više mikroprocesora i odgovarajućih uređaja masovne memorije, kao što su mehuraste memorije, magnetne trake i tvrdi disk, a koji su razvijeni u IBK u Vinči.

Problem je rešavan u više faza. U rešavanju problema učestvovalo je više ljudi, svaki sa određenim specifičnim zadatkom koji ima jasan odnos prema drugim specifičnim zadacima u okviru problema. Arhitektura hardvera je specijalizovana, tako da rešenje sadrži prednosti koje se sreću kod većih računarskih sistema (npr. pored multipleksiranja u vremenu koje obezbeđuje prividni paralelizam izvršenja procesa, postoji i iskorišćenje "I/O wait" vremena jednog procesa za izvršavanje drugih procesa nezavisne strukture). Arhitektura softvera pre svega odražava određene korake u sprovođenju zahteva, zadatih na relativno visokom nivou u niz manjih zahteva nižeg nivoa apstrakcije. Softver je podeljen u module koji, s jedne strane izražavaju podelu funkcija unutar sistema, a s druge strane izražavaju jedinice razvoja softvera koje se pojedinačno relativno lako razvijaju uz jasno definisanu međusobnu komunikaciju i zavisnosti. Modularnost softvera U/I sistema je pre svega potpomognuta nukleusom operativnog sistema koji se koristi u konkretnoj realizaciji (Operating System Processor 80130) i koji obezbeđuje primitive visokog nivoa. Uopšte, modularnost U/I sistema se ističe kao osnovna njegova prednost koja implicira jednostavnost, brz razvoj, lakše održavanje i mogućnost za proširenja

2. ARHITEKTURA HARDVERA

U/I sistem za upravljanje uređajima masovne memorije sadrži u osnovnoj konfiguraciji četiri procesorske komponente. Glavni mikroprocesor je 8086 i on predstavlja integrišući elemenat u sistemu. Takođe, 8086 ostvaruje osnovnu spregu U/I sistema sa ostalim delovima složenog mikroračunarskog sistema. Ostali sistemi složenog mikroračunarskog sistema vide U/I sistema kao skup uređaja visoke inteligencije sa sposobnošću konkurentnog izvršavanja zahteva visokog nivoa apstrakcije. Osnovnu operativnu podršku obezbeđuje

specijalizovana komponenta 80130 koja sadrži nukleus operativnog sistema. Primitivi ovog hardverizovanog nukleusa su kompatibilni sa nukleusom iRMX operativnog sistema, ali ne impliciraju da viši slojevi konkretne realizacije operativnog sistema moraju da budu implementirani na isti način kao i u iRMX sistemu. Osnovnu izvršnu podršku obezbeđuju dva specijalizovana U/I mikroprocesora 8089. Oba U/I mikroprocesora su lokalno spregnuta (Local Mode) sa 8086 i njihov osnovni zadatak je da budu inteligentni DMA kontroleri.



Slika 1

U svakom naprednijem U/I sistemu struktura prekida je od esencijalne važnosti. Prenos upravljanja (kontrola) pomoću prekida je osnovni mehanizam kojim se postiže da asinhroni događaji u sistemu budu uskladjeni. Pre svega misli se na logičku uskladjenost, tj. logika izvršenja zahteva je sinhrona, a radnje koje se obavljaju na komponentama U/I sistema u toku izvršenja zahteva su asinhronne. Na Slici 1. prikazana je arhitektura U/I sistema. Prikaz na slici nije šematski orijentisan, tj. ne prikazuje realne magistrale i veze između komponenta U/I sistema, već je funkcionalno orijentisan, tj. prikazuje osnovne puteve prenosa upravljanja.

Konceptualno, postoje dva nivoa prekida u sistemu. Niži nivo su prekidi koje kontroler uređaja (mehuraste memorije, magnetne trake) šalje U/I procesoru 8089. Viši nivo su prekidi koje U/I procesor šalje osnovnoj procesorskoj strukturi 8086/80130. Svi prekidi za 8086 se kanališu kroz inherentni kontroler prekida komponente 80130. Kontroler prekida ima osam ulaznih linija koje su prioritarno

uredjene, što omogućava fino podešavanje odziva sistema za datu aplikaciju. Jednom prekidu višeg nivoa može da odgovara više prekida nižeg nivoa. Npr. U/I procesor doživljava prekid za svaki preneti bajt, dok osnovni procesor doživljava prekid za svaki preneti blok bajtova. Realno postoji samo viši nivo prekida. Niži nivo prekida je hardverizovan i pojavljuje se u vidu DMA ciklusa između kontrolera uređaja i U/I procesora koji se ponaša kao inteligentni DMA kontroler.

Pre nego što opišemo osnovnu koncepciju U/I sistema i pokažemo gde se javlja "I/O wait" vreme opisaćemo upravljačke programe uređaja (software device drivers).

3. UPRAVLJAČKI PROGRAMI

Upravljački programi su osnovne kontrolne strukture koje se nalaze na najnižem nivou hijerarhije modula U/I sistema i oni obavljaju osnovne radnje na uređaju. Oni su zavisni od vrste uređaja (upravljački program za mehurastu memoriju se bitno razlikuje od upravljačkog programa za magnetnu traku). Upravljački programi se realizuju u dva nivoa koja su uslovljena efikasnošću razvoja sistema. Prvi nivo čine upravljački programi uređaja u nepoznatom programskom okruženju. Njih razvija konstruktor uređaja. Programi su razvijeni u assembleru za 8086 i 8089 mikroprocesora. U njima se ni na koji način ne podrazumeva konkretan operativni sistem ili neka posebna struktura podataka. Drugi nivo razvija sistemski programer. Programi su napisani u višem programskom jeziku PL/M-86 i u njihovoj logici i strukturi podataka nad kojom rade ugrađene su informacije vezane za konkretni operativni sistem. Dakle, drugi nivo su upravljački programi uređaja u poznatom programskom okruženju. Podelom upravljačkih programa na dva nivoa odgovara se na dva pitanja koja se postavljaju pri postavljanju zahteva: šta treba uraditi? (prvi nivo) i u kojoj formi postaviti zahtev? (drugi nivo).

4. ARHITEKTURA U/I SISTEMA I ANALIZA VREMENA ČEKANJA NA IZVRŠENJE ZAHTEVA (I/O WAIT TIME)

Na nivou upravljačkih programa nije moguće osetiti dobitak ostvarenje konfiguracije hardvera. Tek kada se sistem posmatra na nivou procesa koji se odvijaju u njemu može se uočiti da postoje trenuci

kada se u jednom delu sistema (npr. na mehurastoj memoriji) nešto radi za jedan proces (iako je on tad blokiran) dok se istovremeno u drugom delu sistema (na 8086 i 80130) nešto radi za druge procese. To što drugi procesi rade može biti i postavljanje zahteva za U/I radnjom na uređaju koji je upravo zauzet izvršenjem nekog prethodnog zahteva.

Kada 8086 obradi zahtev do određenog nivoa tada pošalje signal ka 8089 preko linije CA (Channel Attention, slika 1). Zbog načina konfigurisanja mikroprocesora 8086 i 8089 (Local Mode) pristup magistrali se dodeli U/I procesoru koji izvrši dalju obradu zahteva. Ovaj deo obrade zahteva je relativno kratak i predstavlja u suštini pripremu za DMA transfer. Za to vreme procesori 8086/ ne mogu raditi na drugom procesu. Kada počne DMA transfer 8089 oslobađa magistralu do početka prvog DMA ciklusa (linija DRQ). Vreme koje protekne do prvog DMA ciklusa je na mehurastoj memoriji prosečno 40 ms. To je veliko vreme za koje 8086/80130 imaju pristup na magistralu i mogu da rade na drugim procesima. Upravo je to vreme gde se ostvaruje stvarni paralelizam u radu sem onog prividnog paralelizma koji podržava komponenta 80130. Posle prvog DMA ciklusa sledeći se javljaju velikom brzinom. Između njih se magistrala takodje oslobađa, ali taj deo vremena nije tako značajan, jer je ukupno vreme prenosa relativno malo. Posle završetka DMA transfera 8089 šalje prekidni signal na 80130, što aktivira poseban proces koji okončava izvršenje U/I zahteva.

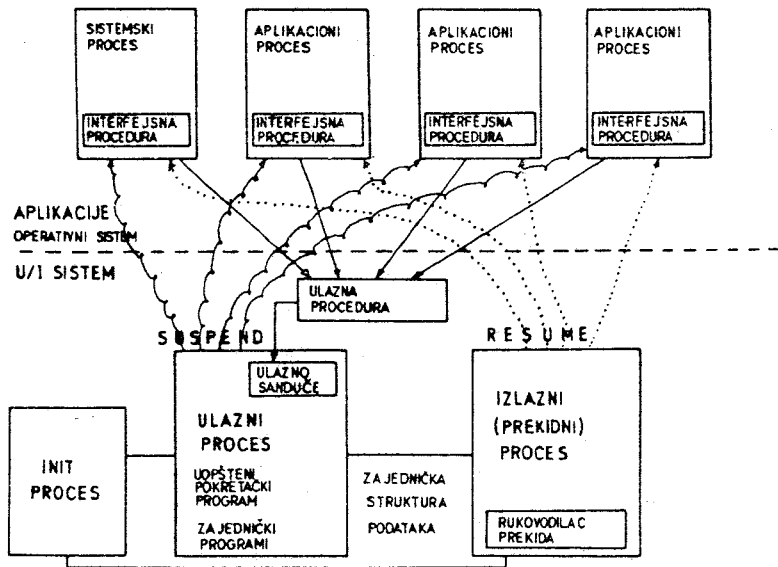
Šta se događa sa procesom koji je izdao U/I zahtev u toku izvršenja tog zahteva? Po definiciji proces je sekvencijalna radnja i dok se njegov zahtev izvršava na uređaju procesor nema šta da radi za taj proces. Potrebno je da proces na neki način suspendovan sa procesora dok je efektivno na uređaju. Ako proces samog sebe suspenduje ne postoji način da se on aktivira po obavljanju zahteva. Proces mora o svom zahtevu da obavesti "jedan poseban proces" višeg prioriteta koji će ga identifikuje, suspenduje i sačuva trag o njemu. Taj poseban proces će pokrenuti upravljačke programe uređaja, ali i omogućiti drugim spremnim procesima da se izvršavaju u "medjuvremenu". Kad se U/I zahtev izvrši "drugi poseban proces" visokog prioriteta na osnovu sačuvanog traga prevešće suspendovani proces u stanje spremno za izvršavanje.

Sve gore podvučene reči imaju posebno značenje i označavaju radnje koje se mogu zahtevati od nukleusa operativnog sistema.

Zadatak U/I sistema je upravo da u sebi sadrži gore pomenute "posebne" procese i odgovarajuću strukturu podataka.

5. PROCESI U/I SISTEMA

Prvi "poseban" proces se naziva ulazni proces U/I sistema (slika 2). On ne poziva direktno upravljačke programe uređaja (drugi nivo), već poziva uopšteni pokretački program uređaja.



Slika 2.

Upravljački programi uređaja (drugi nivo) nisu nezavisni od uređaja iako su prilagođeni programskom okruženju. Za svaku vrstu uređaja (mehurasta memorija, magnetna traka, tvrdi disk) posebno se pišu ovi programi. Za više sličnih vrsta uređaja (kao što su konkretni uređaji upotrebljeni u ovoj implementaciji U/I sistema: Intel mehuraste memorije, Qantex magnetna traka, Priam disk) moguće je napisati uopšteni upravljački program koji obavlja veći deo obrade zahteva, a tek na najnižem nivou poziva specifični upravljački program. Uopšteni upravljački program implementira u sebi

redove čekanja na uređjajima, politike upravljanja tim redovima i odgovarajućim prihvatnicima. Pomoću uopštenog upravljačkog programa i odgovarajuće strukture podataka (tablice gde su smeštene osobine pojedinih uređjaja) postiže se nezavisnost viših slojeva U/I sistema (tj. operativnog sistema i dalje, aplikacija) od vrste uređjaja kojima U/I sistem upravlja. Pod nezavisnošću se podrazumeva uniformni način iskazivanja U/I zahteva.

Kako aplikacioni proces obaveštava sistemski ulazni proces U/I sistema o svom zahtevu? Prvo, neki sinhronizacioni mehanizam mora postojati. Ulazni proces U/I sistema mora imati jedan objekat gde očekuje U/I zahtev. Kada U/I zahteva nema ulazni proces U/I sistema treba da čeka na tom objektu. Taj objekat je poštansko sanduče (mailbox) i u njega aplikacioni proces šalje U/I zahtev po određenom formatu. Istina, to slanje nije tako direktno i aplikacioni proces nije svestan postojanja sandučeta i nema eksplicitan pristup na njega. Pristup je indirektan, preko interfejsne procedure i ulazne procedure U/I sistema. Interfejsna procedura je deo koda aplikacionog procesa koji se dobija iz određene systemske biblioteke i ona preko softverskog prekida aktivira ulaznu proceduru U/I sistema. Ulazna procedura je u takvom kontekstu da "zna" identifikator ulaznog sandučeta tako da mu može poslati poruku, tj. zahtev. I interfejsna procedura i ulazna procedura se izvršavaju u okolini aplikacionog procesa, tj. koriste njegov stek. Ulazni proces je prilagodjen za sinhronu i asinhronu pozive iz aplikacionog procesa, pri čemu su sinhroni pozivi normalan način korišćenja U/I sistema. Drugi "poseban" proces se naziva izlazni (prekidni) proces U/I sistema (slika 2). On se aktivira na pojavu spoljnjeg prekida na liniji komponente 80130 koja je pridružena datom uređjaju. Znači, svaki poseban uređjaj ima poseban izlazni proces. Primetimo da različiti izlazni procesi mogu koristiti identičan kod. Izlazni proces koristi strukturu podataka U/I sistema na osnovu koje se odredi korespondencija između prekida koji se trenutno obrađuje i procesa koji se treba prevesti u spremno stanje. Takođe, potrebno je ostvariti odgovarajuću sinhronizaciju između ulaznog i izlaznog procesa u korišćenju zajedničke strukture podataka (npr. pri uključivanju zahteva u red čekanja na uređjaju i pri isključenju zahteva iz reda).

Opisani U/I sistem je realizovan u jeziku PL/M-86 sa dva procenta koda u assembleru za 8086. Veličina koda (dakle, bez podataka i

steka) je oko 5kB. Prva verzija radi sa uređajem mehuraste memorije kapaciteta 1MB. Za svaki naredni uređaj predviđa se dodatnih 0,5 kB prirasta u veličini ukupno zauzetog prostora. Navedena zauzeća memorije ne uključuju bilo kakve prihvatnike. Razvoj softvera U/I sistema je trajao oko 4 čovek-meseca, ali je to samo orijentaciona mera, jer se razvoj softvera preklapao sa razvojem hardvera.

6. ZAKLJUČAK

U ovom radu dat je prikaz jednog U/I sistema razvijenog u IBK - Vinča za potrebe složenih mikror računarskih sistema. Izložena je arhitektura hardvera, arhitektura i moduli softvera i strategija realizacije i implementacije softvera u konkretnom programskom okruženju. Izložena je primena opštih principa paralelizacije radnji, konkurentnog izvršavanja procesa i multiprogramiranja unutar operativnog sistema. Osnovni pravci dalje aktivnosti će biti pre svega usmereni ka jednom zatvorenom specijalizovanom operativnom sistemu čiji će važan deo biti upravljački sistem, kao i razrada strategije pristupa uređajima i fino podešavanje performansi.

7. LITERATURA

- [1] M. Doroslovački: "Parametri memorijskog sistema sa magnetnim mehurastim memorijama", VI bosanskohercegovački simpozijum iz informatike, Zbornik radova, knjiga 1, saopštenje 151, Jahorina, 1982.
- [2] A. Zavaljevski, G. Živanović, M. Doroslovački: "Dva pristupa primeni magnetnih traka u mikror računarskom sistemu", VII bosanskohercegovački simpozijum iz informatike, Jahorina, 1983.
- [3] A. Zavaljevski, M. Doroslovački, Ž. Milović: "Masovne memorije u višemikroprocesorskom sistemu", VII bosanskohercegovački simpozijum iz informatike, Jahorina, 1984.
- [4] Intel iOSP86 Support Package Reference Manual, 1982.
- [5] Intel Guide to Writing Device Drives for the iRMX86 I/O system, 81.
- [6] Per Brinch Hansen: "The Architecture of Concurrent Programs, Prentice-Hall, Inc., 1977.