

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1730

Datotečni sustav

Luka Milić

Zagreb, lipanj 2011.

svima koji me poznaju i vama u studiju i režiji

SADRŽAJ

1. Uvod	1
2. Opis sklopovlja za pristup diskovima	2
3. Programsko ostvarenje upravljanja diskom	19
4. Ostvarenje datotečnoga sustava	23
5. Primjer rada	31
6. Zaključak	39
Literatura	40

1. Uvod

Svaki iole složeniji današnji operacijski sustav uključuje datotečni sustav. Bez datotečnoga sustava svakim isključivanjem operacijskoga sustava nestao bi sav rad. Datotečni sustav također vodi računa o organiziranju podataka. Primjer popularnoga suvremenog datotečnog sustava jest NTFS. Datotečni sustavi najčešće rukuju tvrdim diskovima. Većina tvrdih diskova spaja se na računalo preko sučelja ATA. SATA je nadogradnja toga sučelja. AHCI je sučelje za programsku komunikaciju s uređajima koji sklopovski komuniciraju s uređajima SATA.

U prvome poglavlju opisuje se sučelje AHCI. U drugome poglavlju opisuje se upotreba sučelja. U trećemu poglavlju opisuje se ostvarenje jednostavnoga datotečnoga sustava. U četvrtome poglavlju opisuje se primjer rada ovoga sustava.

2. Opis sklopovlja za pristup diskovima

U ovome poglavlju opisuje se sučelje AHCI kao izvod iz specifikacije onoga što je potrebno za razumijevanje. Vidjeti [Boyd (2008)], [N/A (2009)] i [Stevens (2007)].

2.1. Uvod

2.1.1. Pregled

Specifikacija AHCI određuje funkcionalno ponašanje i programsko sučelje Advanced Host Controller Interfacea, koji je sklopovski mehanizam koji dopušta programima komunicirati sa uređajima Serial ATA. AHCI je uređaj razreda PCI koji glumi ulogu stroja za pomicanje podataka između sustavske memorije i uređaja Serial ATA.

Domaćinski uređaji AHCI (na koje se poziva kao na domaćinske sabirničke prilagodnike ili HBA) podržavaju od 1 do 32 priključka. HBA mora podržavati uređaje ATA i ATAPI i mora podržavati protokole i PIO i DMA. HBA smije opcionalno podržavati naredbeni popis na svakome priključku za smanjivanje suvišnoga usputnog rada i da podrži Serial ATA Native Command Queueing preko protokola FPDMA Queued Command za svaki uređaj od do 32 unosa. HBA smije opcionalno podržavati 64-bitno adresiranje.

AHCI opisuje sustavsku memorijsku strukturu koja sadrži generičko područje za kontrolu i stanje i tablicu unosa koja opisuje naredbeni popis (HBA koji ne podržava naredbeni popis imat će dubinu od jedan za ovu tablicu). Svaki unos naredbenoga popisa sadržava podatak nužan za programiranje uređaja SATA i pokazivač na opisničku tablicu za prenošenje podataka između sustavske memorije i uređaja.

2.1.2 Teorija operacije

AHCI uzima osnove koncepta raspršivačko-okupljavačkoga popisa Bus Master IDE-a i proširuje ga da smanji procesorov/programski suvišni usputni rad i pruži podršku za

mogućnosti Serial ATA poput dodavanja i zamjene uređaja tijekom rada, upravljanja potrošnjom i pristupanja na nekolicinu uređaja bez izvođenja emulacije *master/slave*.

Komunikacija između uređaja i programa kreće se od zadačne datoteke preko pristupa širokih bajt naredbenome FIS-u smještenomu u sustavskoj memoriji koji HBA dohvaća. Ovo znatno smanjuje vrijeme naredbenoga postavljanja, dopuštajući da se mnogi drugi uređaji dodaju jedincatome domaćinskom kontroleru. Program više ne komunicira izravno s uređajem preko zadačne datoteke.

AHCI je određen da zadrži HBA razmjerno jednostavnim tako da može glumiti ulogu podatakovnoga pomicatelja. Od HBA-a koji implementira AHCI ne zahtijeva se parsiranje ikoje naredbe ATA ili ATAPI kako se prenose na uređaj, iako mu se ne zabranjuje činjenje toga.

Svi podatkovni prijenosi između uređaja i sustavske memorije događaju se kroz HBA koji glumi ulogu sabirničkoga nadgledača u sustavsku memoriju. Bila transakcija tipa DMA ili tipa PIO, HBA dohvaća i skladišti podatke u memoriju, razbremenjujući procesor. Nema pristupljivoga podatkovnog priključka.

Svi se prijenosi izvode upotrebljavajući DMA. Upotreba naredbenoga tipa PIO snažno se obeshrabruje. PIO ima ograničenu podršku za greške - na primjer, završno polje stanja prijenosa PIO daje se HBA-u tijekom PIO Setup FIS-a, prije nego što se prenose podatci. Ipak, neke se naredbe smiju izvoditi samo preko naredbi PIO (poput IDENTIFY DEVICE). Neke HBA-ove implementacije smiju ograničiti podršku PIO na jedan blok podataka DRQ po naredbi.

AHCI određuje normirani mehanizam za implementiranje naredbenoga reda SATA upotrebljavajući DMA Setup FIS. HBA koji podržava čekanje u redu ima pojedinačne uređaje u naredbenome popisu alocirane u sustavskoj memoriji za sve naredbe. Program može smjestiti naredbu u bilo koji prazan utor i obavijestivši HBA preko registarskoga pristupa, HBA će dohvatiti naredbu i prenijeti je. Oznaka koja se vrati u DMA Setup FIS-u upotrebljava se kao indeks u naredbeni popis za pribavljanje raspršivačko-okupljavačkoga popisa upotrebljavanoga u prijenosu.

Ovaj naredbeni popis može upotrebljavati sustavski program i HBA čak i kad se treba prenijeti naredba koja nije stavljena u red. Sustavski program može još uvijek smjestiti višestruke naredbe u popis, bilo DMA, PIO ili ATAPI i HBA će prošetati popisom prenoseći ih.

Ova se višestruka upotreba naredbenoga popisa postiže tako da HBA samo pomiče svoj pokazivač naredbenoga popisa kad uređaj očisti bitove BSY, DRQ i ERR. Sustavski program odgovoran je za osiguranje da se naredbe koje su stavljene u red i naredbe koje nisu stavljene u red ne miješaju u naredbenome popisu za isti uređaj s iznim-

kom naredbe NCQ Unload. NCQ Unload naredba je koja nije stavljena u red koja se može izdati za parkiranje diskovne glave u nuždi. Više podataka o NCQ Unloadu je u odjeljku 5.6.4.3 specifikacije AHCI.

2.2. HBA-ovi konfiguracijski registri

Ovaj odjeljak detaljno opisuje kako bi se trebali konstruirati PCI Header i PCI Capabilities za HBA AHCI. Prikazana polja duplicirana su iz vlastitih specifikacija PCI. Dokumenti PCI normativne su specifikacije za te registre i ovaj odjeljak detaljno opisuje dodatne zahtjeve za HBA AHCI.

Tablica 2.1: HBA-ovi konfiguracijski registri

Početak (hex)	Svršetak (hex)	Naziv
00	3F	Zaglavlje PCI

2.2.1. Zaglavlje PCI

Tablica 2.2: Zaglavlje PCI

Početak (hex)	Svršetak (hex)	Simbol	Naziv
00	03	ID	Identifikatori
00	0B	CC	Razredni kod
24	27	ABAR	Osnovna adresa AHCI

2.2.1.1. Odmak 00h: ID - Identifikatori

Tablica 2.3: Odmak 00h: ID - Identifikatori

Bitovi	Tip	Reset	Opis
31:16	RO	Impl. Spec.	Uređajski ID (DID): pokazuje koji je uređajni broj dodijelio prodavatelj.
15:00	RO	Impl. Spec.	Prodavateljski ID (VID): 16-bitno polje koje pokazuje društvo prodavatelj, koje je dodijelio PCI SIG.

2.2.1.2. Odmak 09h: CC - Razredni kod

Tablica 2.4: Odmak 09h: CC - Razredni kod

Bitovi	Tip	Reset	Opis
23:16	RO	01h	Kod osnovnoga razreda (BCC): Pokazuje da je ovo masovni skladišni uređaj.
15:08	RO	Impl. Spec.	Kod podrazreda (SCC): Kad se postavi na 06h, pokazuje da je ovo uređaj Serial ATA.
07:00	RO	Impl. Spec.	Programsko sučelje (PI): Kad se postavi na 01h i Podrazredni se kod postavi na 06h, pokazuje da je ovo HBA AHCI koji ima glavnu reviziju 1 (kao što je specificirano u registru Inačica AHCI).

Informativna bilješka: Za HBA-ove koji podržavaju RAID, vrijednost reset Podrazrednoga koda treba biti 04h i vrijednost reset Programskoga sučelja treba biti 00h.

2.2.1.3. Odmak 24h: ABAR - Osnovna adresa AHCI

Ovaj registar alocira prostor za HBA-ove memorijske registre određene u odjeljku 3. ABAR se mora alocirati da sadrži dovoljno prostora za globalne registre AHCI, specifične registre priključka za svaki priključak i sav prodavateljski specifični prostor (ako treba). Dopustivo je imati prodavateljski specifičan prostor iza specifičnih registara priključka za zadnji HBA-ov priključak.

Tablica 2.5: Odmak 24h: ABAR - Osnovna adresa AHCI

Bitovi	Tip	Reset	Opis
31:13	RW	0	Osnovna adresa (BA): Osnovna adresa registarskoga memorijskog prostora. Ovo predstavlja memorijski prostor za podršku 32 priključka. Za HBA-ove koji podržavaju manje od 32 priključka, dopušta se da više bitova bude RW i dakle troši se manje memorijskoga prostora. Za HBA-ove koji imaju prodavateljski specifičan prostor na kraju specifičnoga memorijskoga prostora priključka, dopušta se da je više bitova RO tako da se troši više memorijskoga prostora.
12:04	RO	0	<i>Predbilježeno</i>
03	RO	0	Preddohvativo (PF): Pokazuje da ovaj raspon nije preddohvativ.
02:01	RO	00	Tip (TP): Pokazuje da se ovaj raspon može preslikati bilo gdje u 32-bitni memorijski prostor.
00	RO	0	Pokazatelj resursnoga tipa (RTE): Pokazuje zahtjev za registarskim memorijskim prostorom.

2.3. HBA-ovi memorijski registri

Memorijsko preslikani registri unutar HBA-a postoje u memorijskom prostoru koji nije smjestiv u priručnu memoriju. Dodatno, zaključane se adrese ne podržavaju. Ako program pokuša izvesti zaključane transakcije u registre, mogu se zbiti nepredodređeni ishodi. Registarski pristupi imat će najveću veličinu od 64 bita; 64-bitni pristupi ne smiju prijeći 8-bitnu granicu poravnavanja.

Registri su razlomljeni u dva odjeljka - globalni registri i kontrola priključka. Svi registri koji počinju ispod adrese 100h globalni su i namijenjeni za primjenu na čitav HBA. Kontrolni registri priključka isti su za sve priključke i ima toliko mnogo registarskih skupova koliko ima priključaka.

Svi neodređeni registri i svi predbilježeni bitovi unutar registara vraćaju '0' kad se čitaju.

Tablica 2.6: HBA-ovi memorijski registri

Početak	Svršetak	Opis
00h	2Bh	Generička kontrola domaćina
100h	17Fh	Kontrolni registri priključka priključka 0

2.3.1. Generička kontrola domaćina

Sljedeći se registri primjenjuju na čitav HBA.

Tablica 2.7: Generička kontrola domaćina

Početak	Svršetak	Simbol	Opis
0Ch	0Fh	PI	Implementirani priključci

2.3.1.1. Odmak 0Ch: PI - implementirani priključci

Ovaj registar pokazuje koje priključke HBA izlaže. BIOS ga učitava. Pokazuje koji su priključci koje HBA podržava dostupni programu na upotrebu. Na primjer, na HBA-u koji podržava 6 priključaka kao što pokazuje CAP.NP, samo priključci 1 i 3 mogu biti dostupni, s priključcima 0, 2, 4 i 5 nedostupnima.

Program ne smije čitati ili pisati u registre unutar nedostupnih priključaka.

Namjera je ovoga registra dopustiti sustavskim prodavateljima graditi platforme koje podržavaju manje od punoga broja priključaka implementiranih na siliciju HBA-a.

Tablica 2.8: Odmak 0Ch: PI - Implementirani priključci

Bit	Tip	Reset	Opis
31:0	RO	HwInit	Implementirani priključci (PI): Ovaj je registar bitovno znatan. Ako je bit postavljen u '1', odgovarajući je priključak dostupan programu na uporabu. Ako je bit očišćen u '0', priključak nije dostupan programu na uporabu. Najveći broj bitova postavljen u '1' neće premašiti CAP.NP + 1, iako broj bitova postavljenih u ovome registru smije biti manji od CAP.NP + 1. Najmanje jedan bit bit će postavljen u '1'.

2.3.2. Registri priključka (jedan skup po priključku)

Sljedeći registri opisuju registre nužne za implementiranje po priključku; svi priključci imat će isto registarsko preslikavanje. Priključak 0 počinje pri 100h, priključak 1 počinje pri 180h, priključak 2 počinje pri 200h, priključak 3 pri 280h, itd. Algoritam za program za određivanje odmaka jest kako slijedi:

$$\text{Odmak priključka} = 100h + (\text{PI-jev utvrđeni bitovni položaj} * 80h)$$

Tablica 2.9: Registri priključka (jedan skup po priključku)

Početak	Svršetak	Simbol	Opis
00h	03h	PxCLB	Osnovna adresa naredbenoga popisa priključka x
08h	0Bh	PxFB	Osnovna adresa FIS-a priključka x
18h	1Bh	PxCMD	Naredba i stanje priključka x
24h	27h	PxSIG	Potpis priključka x
38h	3Bh	PxCI	Izdavanje naredbe priključka x

2.3.2.1. Odmak 00h: PxCLB - Osnovna adresa naredbenoga popisa priključka x

Tablica 2.10: Odmak 00h: PxCLB - Osnovna adresa naredbenoga popisa priključka x

Bit	Tip	Reset	Opis
31:10	RW	Impl Spec	Osnovna adresa naredbenoga popisa (CLB): Pokazuje 32-bitnu osnovnu fizičku adresu za naredbeni popis za ovaj priključak. Ova se osnova upotrebljava pri dohvatanju naredbi za izvršavanje. Struktura na koju pokazuje ovaj adresni raspon duga je 1K bajtova. Ova adresa mora biti 1K bajtno poravnana kao što pokazuju bitovi 09:00 bivajući samo za čitanje.
09:00	RO	0	<i>Predbilježeno</i>

2.3.2.2. Odmak 08h: PxFB - Osnovna adresa FIS-a priključka x

Tablica 2.11: Odmak 08h: PxFB - Osnovna adresa FIS-a priključka x

Bit	Tip	Reset	Opis
31:08	RW	Impl Spec	Osnovna adresa FIS-a (FB): Pokazuje 32-bitnu baznu fizičku adresu za primljene FIS-ove. Struktura na koju pokazuje ovaj adresni raspon duga je 256 bajtova. Ova adresa mora biti 256-bajtno poravnana kao što pokazuju bitovi 07:00 bivajući samo za čitanje. Kad je FIS-ovsko komutiranje u upotrebi, ova je struktura duga 4 KB i adresa će biti 4 KB-poravnana (pozovite se na odjeljak 9.3.3).
07:00	RO	0	<i>Predbilježeno</i>

2.3.2.3. Odmak 18h: PxCMD - Naredba i stanje priključka x

Tablica 2.12: Odmak 18h: PxCMD - Naredba i stanje priključka x

Bit	Tip	Reset	Opis
15	RO	0	Naredbeni popis pokrenut (CR): Kad je ovaj bit postavljen, pogon DMA naredbenoga popisa za ovaj priključak uključen je. Vidi stroj s konačnim brojem stanja AHCI u odjeljku 5.3.2 za podrobnosti o tome kad HBA postavlja i čisti ovaj bit.
14	RO	0	FIS Receive pokrenut (FR): Kad je postavljen, pogon DMA FIS Receive uključen je. Vidi odjeljak 10.3.2 za podrobnosti o tome kad HBA postavlja i čisti ovaj bit.
04	RW	0	FIS Receive omogućen (FRE): Kad je postavljen, HBA smije stavljati primljene FIS-ove u prijamno područje za FIS-ove na koje pokazuje PxFB (i za 64-bitne HBA-ove, PxFBU). Kad je očišćen, ne prihvaća HBA primljene FIS-ove, osim prvoga registarskoga FIS-a D2H nakon inicijalizacijskoga slijeda i nijedan FIS ne stavlja se u prijamno područje za FIS-ove. Sustavski program ne smije postaviti ovaj bit sve dok se PxFB (PxFBU) ne programiraju valjanim pokazivačem na prijamno područje za FIS-ove i ako program želi pomaknuti osnovu, ovaj bit mora prvo biti očišćen i program mora čekati da se očisti bit FR u ovome registru. Pozovite se na odjeljak 10.3.2 za važna ograničenja na to kad se FRE smije postaviti i očistiti.
00	RW	0	Pokret (ST): Kad je postavljen, HBA smije obrađivati naredbeni popis. Kad je očišćen, HBA ne smije obrađivati naredbeni popis. Kad god se ovaj bit promijeni iz '0' u '1', HBA počine obrađivati naredbeni popis na unosu '0'. Kad god se ovaj bit promijeni iz '1' u '0', HBA očisti registar PxCI nakon što HBA stavi kontroler u stanje praznoga hoda. Ovaj će bit program postaviti u '1' nakon što se PxCMD.FRE postavi u '1'. Pozovite se na odjeljak 10.3.1 za važna ograničenja na to kad se ST može postaviti u '1'.

2.3.2.4. Odmak 24h: PxDIG - Potpis priključka x

Ovo je 32-bitni registar koji sadrži početni potpis spojenoga uređaja kad se prvi D2H Register FIS primi od toga uređaja. Ažurira se jednom nakon resetnoga slijeda.

Tablica 2.13: Odmak 24h: PxDIG - Potpis priključka x

Bit	Tip	Reset	Opis										
31:00	RO	0	Potpis (SIG): Sadrži potpis primljen od uređaja na prvome D2H Register FIS-u. Bitovni je redoslijed kako slijedi: <table border="1"><thead><tr><th>Bit</th><th>Polje</th></tr></thead><tbody><tr><td>31:24</td><td>LBA High Register</td></tr><tr><td>23:16</td><td>LBA Mid Register</td></tr><tr><td>15:08</td><td>LBA Low Register</td></tr><tr><td>07:00</td><td>Sector Count Register</td></tr></tbody></table>	Bit	Polje	31:24	LBA High Register	23:16	LBA Mid Register	15:08	LBA Low Register	07:00	Sector Count Register
Bit	Polje												
31:24	LBA High Register												
23:16	LBA Mid Register												
15:08	LBA Low Register												
07:00	Sector Count Register												

2.3.2.5. Odmak 38h: PxDI - Izdavanje naredbe priključka x

Tablica 2.14: Odmak 38h: PxDI - Izdavanje naredbe priključka x

Bit	Tip	Reset	Opis
31:0	RW1	0	Izdane naredbe (CI): Ovo je polje bitovno znatno. Svaki bit odgovara naredbenome utoru, gdje bit 0 odgovara naredbenome utoru 0. Ovo polje postavlja program kako bi pokazao HBA-u da je naredba izgrađena u sustavskoj memoriji za naredbeni utor i da se može poslati uređaju. Kad HBA primi FIS koji očisti bitove BSY, DRW i ERR za naredbu, očisti odgovarajući bit u ovome registru za taj naredbeni utor. Bitove u ovome polju program će postaviti u '1' samo kad je PxCMD postavljen u '1'. Ovaj se bit također čisti kad program zapiše PxCMD.ST iz '1' u '0'.

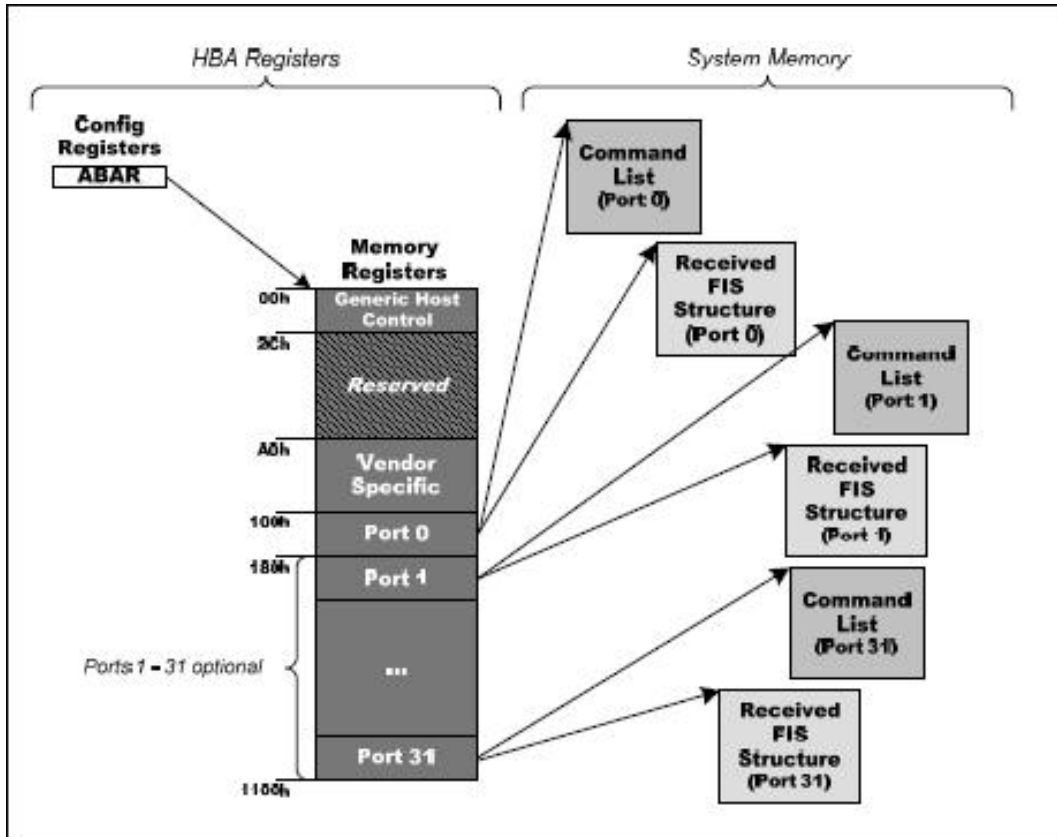
2.4. Sustavske memorijske strukture

2.4.1. HBA-ova upotreba memorijskoga prostora

Većina komunikacije između programa i uređaja SATA jest kroz HBA preko opisnika sustavske memorije, koji pokazuju stanje svih primljenih i poslanih FIS-ova, kao

i pokazivača za podatkovne prijenose. Neka dodatna komunikacija obavlja se preko registara u HBA-u, za svaki priključak i za globalnu kontrolu.

Vizualna podjela HBA-ova memorijskog prostora pokazuje se u Slici 2.1.

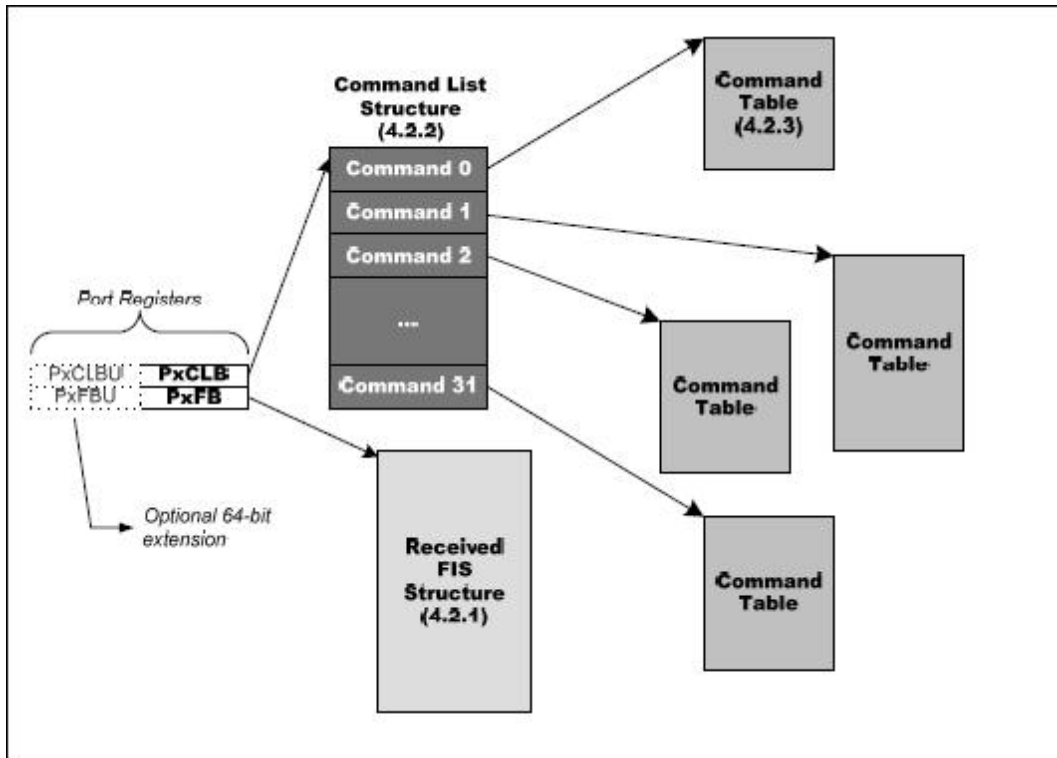


Slika 2.1: HBA-ova upotreba memorijskoga prostora

2.4.2. Upotreba memorije priključka

Ima dva opisnika po priključku koje se upotrebljava za prebacivanje podataka. Jedan je opisnik FIS-a, koji sadržava FIS-ove primljene od uređaja, a drugi je naredbeni popis, koji sadrži popis 1 do 32 naredbe dostupne za priključak na izvršenje.

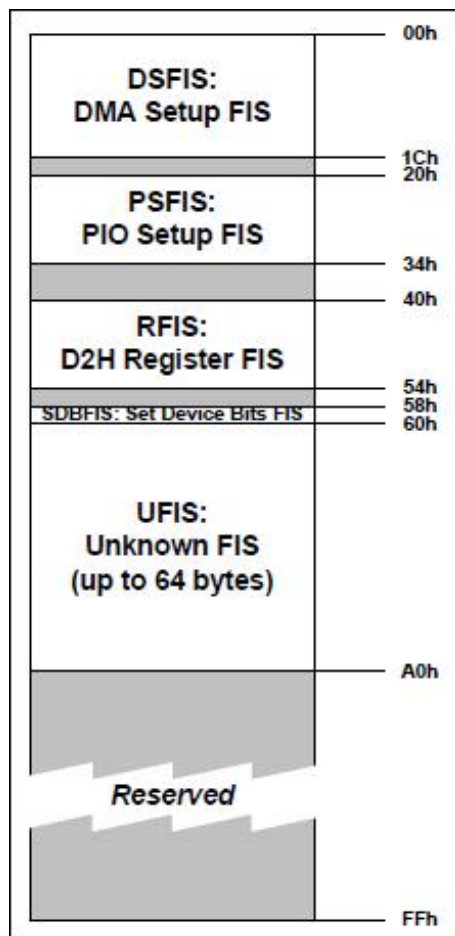
Osnova je za oba pokazivača 64-bitna vrijednost (32 bita za HBA-ove koji ne podržavaju 64-bitno adresiranje). Pregled ukupne strukture pokazuje se u Slici 2.2, a sljedeći odjeljci opisuju svako područje.



Slika 2.2: Sustavske memorijske strukture priključka

2.4.2.1. Struktura Recieved FIS

HBA upotrebljava područje sustavske memorije za komunikaciju podataka o primljenim FIS-ovima. Na ovu strukturu pokazuju PxFBU i PxFB. Struktura se pokazuje u Slici 2.3.



Slika 2.3: Organizacija Received FIS-a

Kad DMA Setup FIS stigne od uređaja, HBA ga kopira u područje DSFIS ove strukture.

Kad PIO Setup FIS stigne od uređaja, HBA ga kopira u područje PSFIS ove strukture.

Kad D2H Register FIS stigne od uređaja, HBA ga kopira u područje RFIS ove strukture.

Kad Set Device Bits FIS stigne od uređaja, HBA ga kopira u područje SDBFIS ove strukture.

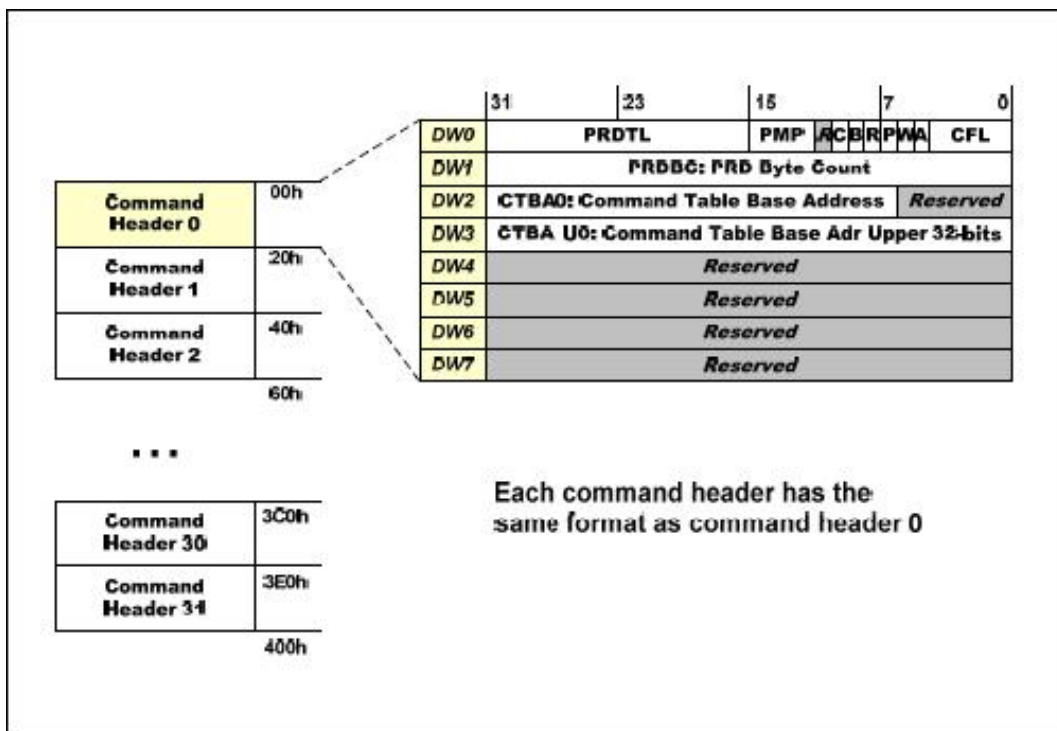
Kad nepoznati FIS stigne od uređaja, HBA ga kopira u područje UFIS u ovoj strukturi i postavi PxSERR.DIAG.F, što se odražava u PxIS.UFS-u kad se FIS postavi u memoriju. Najviše 64 bajtova nepoznatoga tipa FIS-a može se poslati HBA-u. Ako stigne nepoznati FIS koji je duži od 64 bajta, FIS se smatra ilegalnim i rukuje se njim kao što je opisano u odjeljku 6.1.2. Iako je duljina FIS-a HBA-u nepoznata, očekuje se da je zna sustavski program i dakle program će obraditi samo valjane bajtove. Od HBA se ne zahtijeva toleriranje primanja nepoznatoga FIS-a kad HBA očekuje Data FIS od

uređaja ili kad HBA treba prenijeti Data FIS uređaju prema naredbenome protokolu koji se upotrebljava.

HBA će poduzeti radnje opisane u 5.3 kad je primljen FIS, što uključuje ažuriranje strukture Received FIS kao što je prije ocrtno, stvarajući prekid ako je prikladno, itd.

2.4.2.2. Struktura naredbenoga popisa

Slika 2.4 pokazuje strukturu naredbenoga popisa. Svaki unos sadržava naredbeno zaglavlje, koje je 32-bajtna struktura koja detaljno opisuje smjer, tip i raspršivačko-okupljavački pokazivač naredbe. Daljnje detaljnosti svakoga polja popisane su ispod.



Slika 2.4: Struktura naredbenoga popisa

Polja u naredbenome zaglavlju su:

Tablica 2.15: DW 0 - Opisna informacija

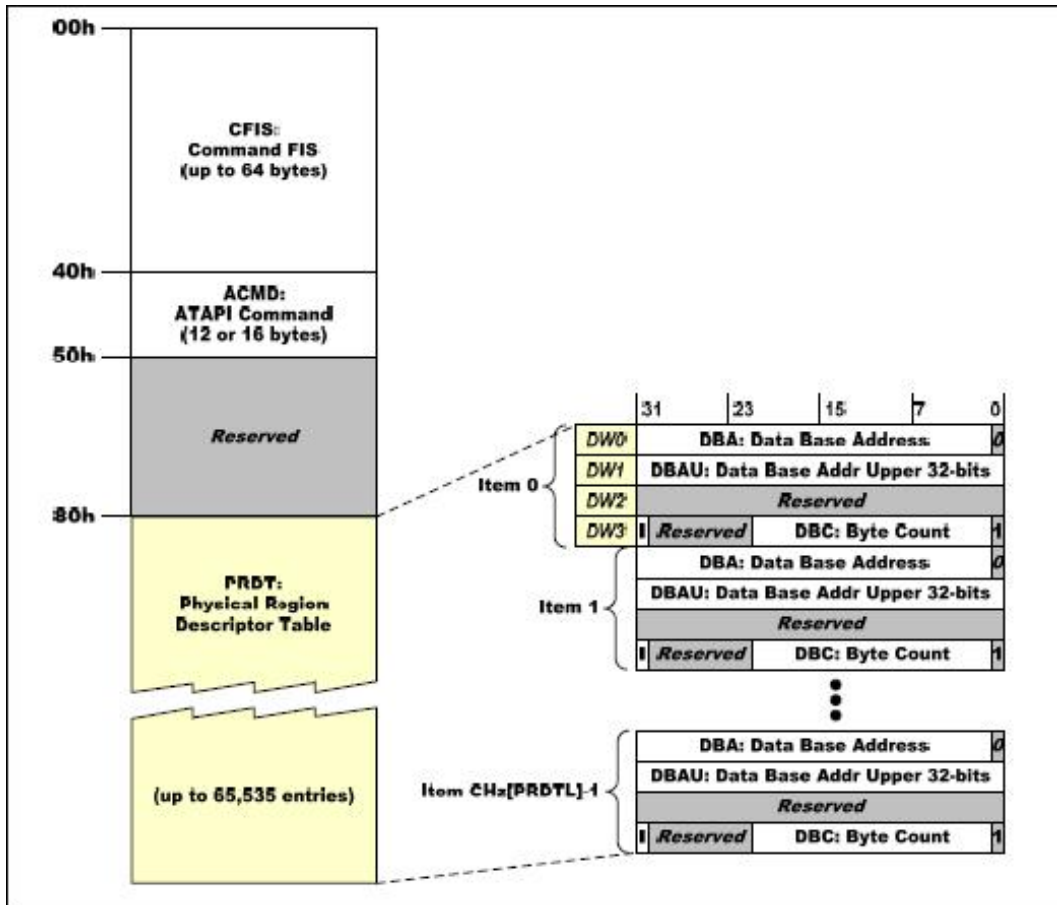
Bit	Opis
31:16	Duljina opisničke tablice fizičke regije (PRDTL): Duljina tablice raspršivačko-okupljivačkih opisnika u unosima, nazvana Physical Region Descriptor Table. Svaki je unos 4 DW. '0' predstavlja 0 unosa, FFFFh predstavlja 65535 unosa. HBA upotrebljava ovo polje da zna kad prestati dohvaćati PRD-ove. Ako je ovo polje '0', onda se neće zbiti nikakav podatkovni prijenos naredbom.
06	Piši (W): Kad je postavljen, pokazuje da je smjer uređajsko pisanje (podatci iz sustavske memorije u uređaj). Kad je očišćen, pokazuje da je smjer uređajsko čitanje (podatci iz uređaja u sustavsku memoriju). Ako je ovaj bit postavljen i postavljen je bit P, HBA smije preddohvatiti podatke u iščekivanju primanja DMA Setup FIS-a, DMA Activate FIS-a ili PIO Setup FIS-a, kao i preddohvaćati PRD-ove.
04:00	Duljina naredbenoga FIS-a (CFL). '0' predstavlja 0 DW, '4' predstavlja 4 DW. Duljina '0' ili '1' nije dopuštena. Najveća vrijednost dopuštena je 10h ili 16 DW. HBA upotrebljava ovo polje da zna duljinu FIS-a koji će poslati uređaju.

Tablica 2.16: DW 2 - Osnovna adresa naredbene tablice

Bit	Opis
31:07	Osnovna adresa opisnika naredbene tablice (CTBA): Pokazuje 32-bitnu fizičku adresu naredbene tablice, koja sadrži naredbeni FIS, ATAPI Command i tablicu PRD. Ova adresa mora biti poravnana na 128-bajtnu liniju priručne memorije, što pokazuju bitovi 06:00 bivajući predbilježenima.
06:00	<i>Predbilježeno</i>

2.4.2.2. Naredbena tablica

Svaki unos u naredbenome popisu pokazuje na strukturu zvanu naredbenom tablicom.



Slika 2.5: Naredbena tablica

Svaka naredba sadržava nekolicinu polja. Polja su podijeljena kako slijedi:

2.4.2.3.1. Naredbeni FIS (CFIS)

Ovo je programsko konstruirani FIS. Za operacije podatkovnoga prijenosa, ovo je format H2D Register FIS kao što je specificiran u specifikaciji Serial ATA Revision 2.6. Ako je spojen Port Multiplier, ovo polje mora imati broj priključka Port Multipliera u samome FIS-u - neće ga dodati HBA. Valjane dužine CFIS-a su 2 do 26 *dwordova* i moraju biti u *dwordovskoj* zrnatosti.

2.4.2.3.2. Naredba ATAPI (ACMD)

Ovo je programsko konstruirano područje od 12 ili 16 bajtova po duljini koje sadrži naredbu ATAPI za prijenos ako je postavljen bit "A" u naredbenome zaglavlju. Naredba ATAPI mora biti ili 12 ili 16 bajtova po duljini. Duljina koju HBA prenese predodređena je PIO Setup FIS-om koji pošalje uređaj zahtijevajući naredbu ATAPI.

2.4.2.3.3. Opisnička tablica fizičke regije (PRDT)

Ova tablica sadrži raspršivačko-okupljavački popis za podatkovni prijenos. Sadržava popis 0 (nema podataka za prijenos) do sve do 65,536 unosa. Podjela svakoga polja u tablici PRD pokazuje se ispod. Stavka 0 poziva se na prvi unos u tablici PRD. Stavka „CH[PRDTL] - 1“ poziva se na zadnji unos u tablici, gdje duljinsko polje dolazi iz polja PRDTL u unosu naredbenoga popisa za ovaj naredbeni utor.

Tablica 2.17: DW 0 - Osnovna adresa podataka

Bit	Opis
31:01	Osnovna adresa podataka (DBA): Pokazuje 32-bitnu fizičku adresu podatkovnoga bloka. Blok mora biti <i>wordovski</i> poravnan, što pokazuje bit 0 bivajući predbilježenim.
00	<i>Predbilježeno</i>

Tablica 2.18: DW 3 - Opisna informacija

Bit	Opis
21:00	Brojnost podatkovnih bajtova (DBC): '0'-težinska vrijednost koja pokazuje duljinu, u bajtovima, podatkovnoga bloka. Najviše 4 MB može postojati za bilo koji unos. Bit '0' ovoga polja mora uvijek biti '1' da pokaže parnu brojnost bajtova. Vrijednost '1' pokazuje 2 bajta, '3' pokazuje 4 bajta, itd.

3. Programsko ostvarenje upravljanja diskom

U ovome poglavlju opisuje se upotreba sučelja AHCI za upravljanje diskom kao niz koraka koje treba poduzeti. Vidjeti [Boyd (2008)], [N/A (2009)] i [Stevens (2007)].

3.1. Nalazak kontrolera

Za nalazak kontrolera mora se ići po svim brojevima od 0 do 65535. Na priključak 0xCF8 zapisuje se 0x80000000 s trenutnim brojem postavljenim na bitove 23:08. Ako s priključka 0xCFC ne dođe 0xFFFFFFFF, na priključak 0xCF8 zapisuje se 0x80000008 s trenutnim brojem postavljenim na bitove 23:08 i ako s priključka 0xCFC dođe 0x10601 na donja 24 bita, prekidamo petlju. Ako smo ispitali sve brojeve, kontrolera AHCI nema na sabirnici. S priključka 0xCFC čita se osnovna adresa.

3.2. Nalazak pogona

Za nalazak pogona mora se ići po svim bitovima od 0 do 31. U polju PI polja GHC ispituje se je li trenutni bit postavljen. Ako jest, ispituje se je li u polju PxsIG odgovarajućega priključka 0x101 i ako jest, prekidamo petlju. Ako smo ispitali sve bitove, pogona SATA nema na kontroleru. Inače, našli smo ga.

3.3. Inicijalizacija priključka

U polje PxCLB našega priključka zapiše se adresa naredbenoga popisa. U polje PxFB našega priključka zapiše se adresa Recieved FIS-a. Isprazni se Recieved FIS.

3.4. Inicijalizacija naredbenoga popisa

Isprazni se naredbeni popis. U polje PRDTL naredbe 0 našega naredbenoga popisa zapiše se 1. U polje CFL naredbe 0 našega naredbenoga popisa zapiše se količnik

veličine CFIS-a i veličine *dworda*. Isprazni se naredbena tablica za čitanje. Njezina se adresa zapisuje u polje CTBA0 naredbe 0 našega naredbenoga popisa. U polje PRDTL naredbe 1 našega naredbenoga popisa zapisuje se 1. U polje W naredbe 1 našega naredbenoga popisa zapisuje se 1. U polje CFL naredbe 1 našega naredbenoga popisa zapiše se količnik veličine CFIS-a i veličine *dworda*. Isprazni se naredbena tablica za pisanje. Njezina se adresa zapisuje u polje CTBA0 naredbe 1 našega naredbenoga popisa.

3.5. Pokretanje sustava

Postave se bitovi 4 i 0 u registru PxCMD našega priključka.

3.6. Čekanje pokretanja

Čeka se dok se bitovi 15 i 14 u registru PxCMD našega priključka ne postave.

3.7. Slanje IDENTIFY DEVICE-a

U polje Command polja CFIS naše naredbene tablice za čitanje zapiše se 0xEC. U polje C polja CFIS naše naredbene tablice za čitanje zapiše se 1. U polje FIS Type polja CFIS naše naredbene tablice za čitanje zapiše se 0x27. U polje DBC polja 0 tablice PRDT naše naredbene tablice za čitanje zapiše se 511. U polje DBA polja 0 tablice PRDT naše naredbene tablice za čitanje zapiše se adresa međuspremnika. Bit 0 polja CI našega priključka postavi se. Čeka se dok se ne očisti.

3.8. Provjeravanje podržava li LBA

Provjerava se je li postavljen bit 25 *dworda* 24 povratne vrijednosti naredbe IDENTIFY DEVICE.

3.9. Provjeravanje podržava li 48-bitni LBA

Provjerava se je li postavljen bit 10 *dworda* 43 povratne vrijednosti naredbe IDENTIFY DEVICE.

3.10. Čitanje broja sektora

Čita se *dword* 30 povratne vrijednosti naredbe IDENTIFY DEVICE.

3.11. Čitanje veličine sektora

Zbroji se *dword* 58 logički posmaknut udesno za 16 bitova i *dword* 59 logički posmaknut ulijevo za 16 bitova. Međuishod se logički posmakne ulijevo za jedan bit. Ako je konačni ishod 0, promijeni se u 512.

3.12. Inicijalizacija naredbe za čitanje

U polje Command polja CFIS naše naredbene tablice za čitanje zapisuje se 0x25. U polje Counter polja CFIS naše naredbene tablice za čitanje zapisuje se 1. U polje Device polja CFIS naše naredbene tablice za čitanje zapisuje se 0x40. U polje DBC polja 0 tablice PRDT naše naredbene tablice za čitanje zapisuje se veličina sektora - 1.

3.13. Inicijalizacija naredbe za pisanje

U polje Command polja CFIS naše naredbene tablice za pisanje zapisuje se 0x35. U polje C polja CFIS naše naredbene tablice za pisanje zapisuje se 1. U polje FIS Type polja CFIS naše naredbene tablice za pisanje zapisuje se 0x27. U polje Counter polja CFIS naše naredbene tablice za pisanje zapisuje se 1. U polje Device polja CFIS naše naredbene tablice za pisanje zapisuje se 0x40. U polje DBC polja 0 tablice PRDT naše naredbene tablice za pisanje zapisuje se veličina sektora - 1.

3.14. Čitanje određenoga sektora s diska

U polje DBA polja 0 tablice PRDT naše naredbene tablice za čitanje zapisuje se adresa gdje će se podatci pročitati. U polje LBA polja CFIS naše naredbene tablice za čitanje zapisuju se donja 24 bita adrese LBA. U polje LBA2 polja CFIS naše naredbene tablice za čitanje zapisuje se gornjih 8 bitova adrese LBA. Postavi se bit 0 registra PxCI našega priključka. Čeka se dok se ne očisti.

3.15. Pisanje određenoga sektora na disk

U polje DBA polja 0 tablice PRDT naše naredbene tablice za pisanje zapisuje se adresa odakle će se podatci zapisati. U polje LBA polja CFIS naše naredbene tablice za pisanje zapisuju se donja 24 bita adrese LBA. U polje LBA2 polja CFIS naše naredbene tablice za pisanje zapisuje se gornjih 8 bitova adrese LBA. Postavi se bit 1 registra PxCI našega priključka. Čeka se dok se ne očisti.

3.16. Zaustavljanje sustava

Očisti se bitove 4 i 0 u registru PxCMD našega priključka.

3.17. Čekanje zaustavljanja

Čeka se dok se bitovi 15 i 14 u registru PxCMD našega priključka ne očiste.

4. Ostvarenje datotečnoga sustava

U ovome poglavlju opisuje se ostvarenje jednostavnoga datotečnoga sustava općenito, jedna po jedna podatkovna struktura i jedan po jedan algoritam.

4.1. Općenito

U ovome je projektu napravljen datotečni sustav za jednostavan operacijski sustav oszur2010/k5. Sučelje ostvarenoga datotečnog sustava sadržava funkcije za formatiranje, formatiranje na brzinu, paljenje, gašenje, stvaranje/ brisanje/ kopiranje/ premještanje/ preimenovanje/ (dohvaćanje atributa) datoteke/direktorija i pandane funkcijama fopen, freopen, fclose, fread, fwrite, fseek, rewind i ftell. Implementacija sustava ima opisnike datoteka, opisnike direktorija i FAT u prvim sektorima diska, koji će se pri paljenju čitati i pri gašenju pisati, a osim njih u memoriji biti će popis (*list*) pandana C-ovim FILE-ovima i opisnika memorijskoga prostora koji datotečni sustav rabi. Ostvareno je i čitanje određenoga sektora s diska i pisanje određenoga sektora na disk kroz komunikaciju s kontrolerom.

4.2. Datotečni opisnici

1. član: Naziv datoteke. Maks. 31 znak. Ne smije sadržavati stazni separator '\'. Znakovi '/', ':', '*', '?', '\', '<', '>' i '|' dopušteni su. Iako i datoteka i direktorij imaju ovu varijablu, ne ide u atribute jer se atributi ne mogu mijenjati. Naziv svih datoteka u naddirektoriju jedinstven je. Nastavak (npr. exe) ne glumi nikakvu ulogu. Sve se trpa u jednu varijablu.

2. član: Identifikator naddirektorija. Redni broj direktorijskoga opisnika. Ne može biti pokazivač jer se svi opisnici čitaju s diska i pišu na disk.

3. član: Atributi datoteke. Nepromjenjivo za sve vanjske funkcije.

4. član: Identifikator sektora prvoga bloka.

5. član: Je li datotečni opisnik zauzet. Je li u ovome bloku memorije datotečni opisnik. Funkcije formatiraj i formatiraj_na_brzinu osigurat će da je početna vrijednost

varijable zauzet kod svih opisnika 0.

4.3. Direktorijski opisnici

1. član: Naziv direktorija. Max 31 znak. Ne smije sadržavati stazni separator '\'. Znakovi '/', ':', '*', '?', '\', '<', '>' i '|' dopušteni su. Iako i datoteka i direktorij imaju ovu varijablu, ne ide u attribute jer se atributi ne mogu mijenjati. Naziv svih direktorija u naddirektoriju jedinstven je.

2. član: Identifikator naddirektorija. Redni broj direktorijskoga opisnika. Ne može biti pokazivač jer se svi opisnici čitaju s diska i pišu na disk. (Ako se radi o vršnome direktoriju, -1 je.)

3. član: Atributi direktorija. Nepromjenjivo za sve vanjske funkcije.

4. član: Prva datoteka u direktoriju. Redni broj datotečnoga opisnika. Ne može biti pokazivač jer se svi opisnici čitaju s diska i pišu na disk. (Ovdje će to biti prva datoteka po abecedi.)

5. član: Prvi direktorij u direktoriju. Redni broj direktorijskoga opisnika. Ne može biti pokazivač jer se svi opisnici čitaju s diska i pišu na disk. (Ovdje će to biti prvi direktorij po abecedi.)

6. član: Sljedeći direktorij u naddirektoriju. Redni broj direktorijskoga opisnika. Ne može biti pokazivač jer se svi opisnici čitaju s diska i pišu na disk. (Ako se radi o vršnome direktoriju, -1 je.) Direktoriji su poredani abecedno (u obliku linearnoga popisa). Glava je popisa varijabla prvi_dir_u_dir naddirektorija.

7. član: Je li direktorijski opisnik zauzet. Je li u ovome bloku memorije direktorijski opisnik. Funkcije formatiraj i formatiraj_na_brzinu osigurat će da je početna vrijednost varijable zauzet kod svih opisnika 0.

4.4. Zapisi u FAT-u

-1 ako je sektor zauzet i ovo je posljednji sektor u popisu, 0 ako je sektor slobodan ili <broj> ako je sektor zauzet i <broj> je sljedeći sektor u popisu.

4.5. Opisnici u popisu otvorenih datoteka

1. član: Pokazivač koji služi za povezivanje instancija strukture u (linearan) popis.

2. član: Pokazivač na trenutno mjesto rada. To je mjesto u bloku na koji pokazuje varijabla osnova. Na M\$-ovu bi kompajleru ovo bilo char *_ptr;

3. član: Pokazivač na trenutni blok (odnosno sektor). U svakoj instanciji strukture odgovara varijabli blok u jednoj instanciji strukture zaglavlje. Na M\$-ovu bi kompajleru ovo bilo char *_base;

4. član: Način otvaranja. Ograničava upotrebu funkcija citaj_iz_dat i pisi_u_dat. Na M\$-ovu bi kompajleru ovo bilo nešto slično int _flag;

5. član: Pokazivač na datotečni opisnik. Na M\$-ovu bi kompajleru ovo bilo nešto slično int _file;

6. član: Koliko je bajtova dodano. Služi za promjenu atributa datoteke pri zatvaranju. Služi i tijekom čitanja i pisanja u datoteku.

7. član: Koliko je bajtova dodano na disku. Služi za promjenu atributa datoteke pri zatvaranju. Varijabla nije jako korisna.

4.6. Popis zaglavlja blokova otvorenih datoteka

1. član: Pokazivač koji služi za povezivanje instancija strukture u (linearni) popis.

2. član: Identifikator sektora.

3. član: Blok (odnosno sektor). U nekim instancijama struktura odgovara varijabli osnova u jednoj instanciji strukture datoteka.

4. član: Je li blok prljav.

4.7. Otvaranje datoteke

Ako je zadan jedan od šest binarnih načina otvaranja datoteke, ovisno o načinu, primjenjuje se odgovarajuće ponašanje fopen. Inače, javi se greška. Stvori se i, inicijalizira i doda u popis opisnik otvorene datoteke. Svi se blokovi datoteke stvore, pročitaju s diska i dodaju u popis zaglavlja blokova.

4.8. Ponovno otvaranje datoteke

Zove se zatvaranje datoteke. Zove se otvaranje datoteke.

4.9. Zatvaranje datoteke

Promijene se atributi datoteke i svih naddirektorija. Uklanjaju se blokovi iz popisa i uništavaju, prljavi među njima zapisuju se na disk. Uklanja opisnik otvorene datoteke i uništava.

4.10. Čitanje iz datoteke

Ako je datoteka otvorena u načinu "wb" ili "ab", javi se greška. Broj bajtova koji treba pročitati postavi se na umnožak parametara veličina i broj. Ako se ne može pročitati toliko, smanji se na onoliko koliko se može pročitati. Ako je broj manji od onoliko koliko je do kraja bloka, to je jednostavan slučaj. Inače se čita do kraja bloka i oduzme koliko je pročitano. Onda se čitaju cijeli blokovi i oduzima veličina bloka sve dok se ne dođe do jednostavnoga slučaja koji je već obrađen.

4.11. Pisanje u datoteku

Ako je datoteka otvorena u načinu "rb", javi se greška. Ako je u načinu "ab" ili "a+b", premota se na kraj. Broj bajtova koji treba pročitati postavi se na umnožak parametara veličina i broj. Najprije se piše do kraja datoteke. Ako je broj manji od onoliko koliko je do kraja bloka, to je jednostavan slučaj. Inače se piše do kraja bloka i oduzme koliko je zapisano. Onda se pišu cijeli blokovi i oduzima veličina bloka sve dok se ne dođe do jednostavnoga slučaja koji je već obrađen. Kad se dođe do kraja datoteke, za svaki prijelaz bloka traže se slobodni zapis u FAT-u i ako postoji, zauzme se i doda se opisnik u popis blokova, a ako nema više slobodnih sektora na disku, zaustavlja se.

4.12. Govorenje položaja

Skaćući po FAT-u broje se blokovi dok se ne nađe onaj na koji pokazuje pokazivač osnovnoga položaja u opisniku otvorene datoteke. Konačni broj blokova pomnoži se s veličinom sektora i doda se razlika pokazivača trenutnoga položaja i pokazivača osnovnoga položaja.

4.13. Traženje položaja

Ovisno o tome odakle se traži, ne doda se ništa, doda se ftell ili se doda veličina datoteke na odmak. Skaćući po FAT-u odbrojavaju se blokovi i oduzima se veličina bloka od odmaka dok ne postane manja od veličine bloka. Pokazivač osnovnoga položaja postavi se na blok. Pokazivač trenutnoga položaja postavi se na blok i uveća za ostatak.

4.14. Premotavanje

Premotavanje je samo zamotano traženje položaja od početka.

4.15. Stvaranje datoteke

Traži se naddirektorij tražene datoteke u direktorijskim opisnicima. Ako ne postoji, javlja se greška. Ako u njemu već postoji isto nazvana datoteka, javlja se greška. Traži se slobodni sektor u FAT-u. Ako više nema slobodnih sektora na disku, javlja se greška. Traži se slobodni opisnik u datotečnim opisnicima. Ako više nema mjesta za novi opisnik, javlja se greška. Zauzme se blok u FAT-u. Zauzme se, zapiše i poveže datotečni opisnik. Promijene se atributi u naddirektorijima.

4.16. Brisanje datoteke

Traži se datoteka u datotečnim opisnicima. Ako ne postoji, javi se greška. Traži se datoteka u opisnicima otvorenih datoteka. Ako postoji, javi se greška. Oslobode se sektori datoteke u FAT-u. Datotečni se opisnik razveže i oslobodi. Promijene se atributi u naddirektorijima.

4.17. Stvaranje direktorija

Traži se naddirektorij traženoga direktorija u direktorijskim opisnicima. Ako ne postoji, javlja se greška. Ako u njemu već postoji isto nazvan direktorij, javlja se greška. Traži se slobodni opisnik u direktorijskim opisnicima. Ako više nema mjesta za novi opisnik, javlja se greška. Zauzme se, zapiše i poveže direktorijski opisnik. Promijene se atributi u naddirektorijima.

4.18. Brisanje direktorija

Traži se direktorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Ako je vršni, javi se greška. Ide se u rekurziju. U rekurziji se oslobodi opisnik. Najprije se ide po svim datotekama u direktoriju. Traži se datoteka u opisnicima otvorenih datoteka. Ako postoji, ponovno se zauzme opisnik i preskoči se datoteka. Oslobode se sektori datoteke u FAT-u. Datotečni se opisnik razveže i oslobodi. Promijene se atributi u naddirektorijima. Onda se ide po svim direktorijima u direktoriju. Opet se ide u istu rekurziju za svaki od njih. Kad se vrati iz nje, razveže se opisnik toga direktorija i promijene atributi u naddirektorijima. Kad se vrati iz početne rekurzije, razveže se opisnik početnoga direktorija i promijene atributi u naddirektorijima.

4.19. Preimenovanje datoteke

Nađe se datoteka u datotečnim opisnicima. Ako ne postoji, javi se greška. Prepiše se naziv datoteke. Mijenjaju se atributi ovdje i u naddirektorijima.

4.20. Dohvaćanje atributa datoteke

Nađe se datoteka u datotečnim opisnicima. Ako ne postoji, javi se greška. Kopira se polje atributa. Mijenjaju se atributi ovdje i u naddirektorijima.

4.21. Preimenovanje direktorija

Nađe se direktorij u datotečnim opisnicima. Ako ne postoji, javi se greška. Prepiše se naziv direktorija. Mijenjaju se atributi ovdje i u naddirektorijima.

4.22. Dohvaćanje atributa direktorija

Nađe se direktorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Kopira se polje atributa. Mijenjaju se atributi ovdje i u naddirektorijima.

4.23. Premještanje datoteke

Traži se stari naddirektorij tražene datoteke u direktorijskim opisnicima. Ako ne postoji, javi se greška. Traži se datoteka u datotečnim opisnicima. Ako ne postoji, javi se greška. Traži se novi naddirektorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Ako u njemu već postoji isto nazvana datoteka, javi se greška. Datotečni se opisnik razveže u jednome, a poveže u drugome direktoriju. Promijene se atributi u jednome i drugome naddirektoriju.

4.24. Kopiranje datoteke

Traži se stari naddirektorij tražene datoteke u direktorijskim opisnicima. Ako ne postoji, javi se greška. Traži se datoteka u datotečnim opisnicima. Ako ne postoji, javi se greška. Traži se novi naddirektorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Ako u njemu već postoji isto nazvana datoteka, javi se greška. Traži se datoteka u opisnicima otvorenih datoteka. Ako postoji, javi se greška. Traži se slobodni opisnik u datotečnim opisnicima. Ako više nema mjesta za novi opisnik, javlja se greška. Ide se po sektorima datoteke i traži se slobodni sektor za svaki. Ako ga nema, javi se greška. Sektor po sektor se kopira. Zauzme se, zapiše i poveže

datotečni opisnik u odredišnome naddirektoriju. Promijene se atributi u jednome i drugome naddirektoriju.

4.25. Premještanje direktorija

Traži se stari naddirektorij traženoga direktorija u direktorijskim opisnicima. Ako ne postoji, javi se greška. Traži se direktorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Traži se novi naddirektorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Ako u njemu već postoji isto nazvan direktorij, javi se greška. Direktorijski se opisnik razveže u jednome, a poveže u drugome direktoriju. Promijene se atributi u jednome i drugome naddirektoriju.

4.26. Kopiranje direktorija

Traži se stari naddirektorij traženoga direktorija u direktorijskim opisnicima. Ako ne postoji, javi se greška. Traži se direktorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Ako je vršni, javi se greška. Traži se novi naddirektorij u direktorijskim opisnicima. Ako ne postoji, javi se greška. Ako u njemu već postoji isto nazvan direktorij, javi se greška. Traži se slobodni opisnik u direktorijskim opisnicima. Ako više nema mjesta za novi opisnik, javlja se greška. Zauzme se, zapiše i poveže direktorijski opisnik u odredišnome naddirektoriju. Promijene se atributi u jednim i drugim naddirektorijima. Ide se u rekurziju. U rekurziji se najprije ide po svim datotekama u direktoriju. Traži se datoteka u opisnicima otvorenih datoteka. Ako postoji, preskoči se. Traži se slobodni opisnik u datotečnim opisnicima. Ako više nema mjesta za novi opisnik, zaustavlja se kopiranje. Ide se po sektorima datoteke i traži se slobodni sektor za svaki. Ako ga nema, zaustavlja se kopiranje. Sektor po sektor se kopira. Zauzme se, zapiše i poveže datotečni opisnik u odredišnome naddirektoriju. Promijene se atributi u jednome i drugome naddirektoriju. Onda se ide po svim direktorijima u direktoriju. Traži se slobodni opisnik u direktorijskim opisnicima. Ako više nema mjesta za novi opisnik, prekida se kopiranje. Zauzme se, zapiše i poveže direktorijski opisnik. Opet se ide u istu rekurziju za svaki od njih.

4.27. Formatiranje

Cijeli se disk prepíše nulama. U nultome direktorijskom opisniku zauzme se i poveže opisnik vršnoga direktorija s proslijeđenim nazivom i završnom dvotočkom. Zapiše se na disk. U prvim zapisima FAT-a zauzme se mjesto za datotečne opisnike, direktorijske opisnike i FAT.

4.28. Formatiranje na brzinu

Datotečni opisnici, direktorijski opisnici i FAT na disku prepisu se nulama. U nul-tome direktorijskom opisniku zauzme se i poveže opisnik vršnoga direktorija s proslijeđenim nazivom i završnom dvotočkom. Zapiše se na disk. U prvim zapisima FAT-a zauzme se mjesto za datotečne opisnike, direktorijske opisnike i FAT.

4.29. Pokretanje

Inicijaliziraju se varijable. Stvaraju se datotečni opisnici. Čitaju se s diska. Stvaraju se direktorijski opisnici. Čitaju se s diska. Svara se FAT. Čita se se diska. Inicijalizira se popis otvorenih datoteka. Inicijalizira se popis zaglavlja blokova.

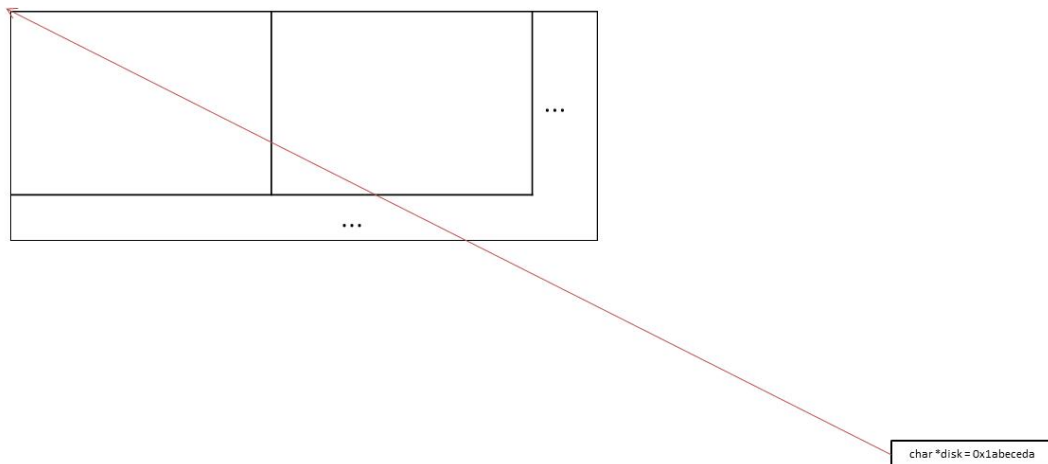
4.30. Zaustavljanje

Datotečni opisnici pišu se na disk. Uništavaju se. Direktorijski opisnici pišu se na disk. Uništavaju se. FAT se piše na disk. Uništava se. Popis otvorenih datoteka uništava se. Popis zaglavlja blokova uništava se.

5. Primjer rada

U ovome poglavlju opisuje se primjer rada ovoga sustava na jednostavnome primjeru čitanja i pisanja u datoteku kao niz naredbi koje se izvršavaju.

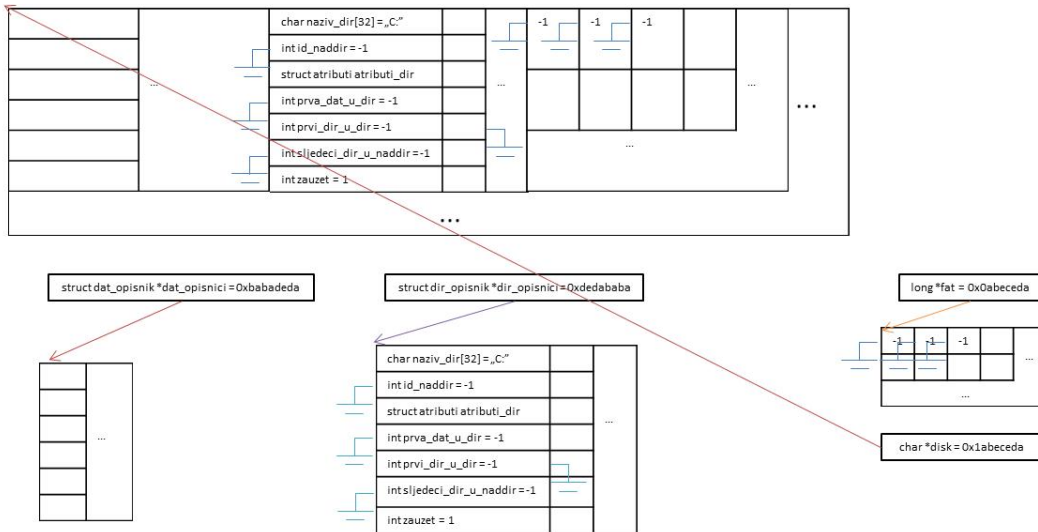
5.0. Početno stanje



Slika 5.1: Početno stanje

Gore lijevo je disk. Disk je podijeljen na sektore. Prikazana su samo dva sektora. Točkice pokazuju da ih ima još. Dolje desno je pokazivač. On pokazuje na disk.

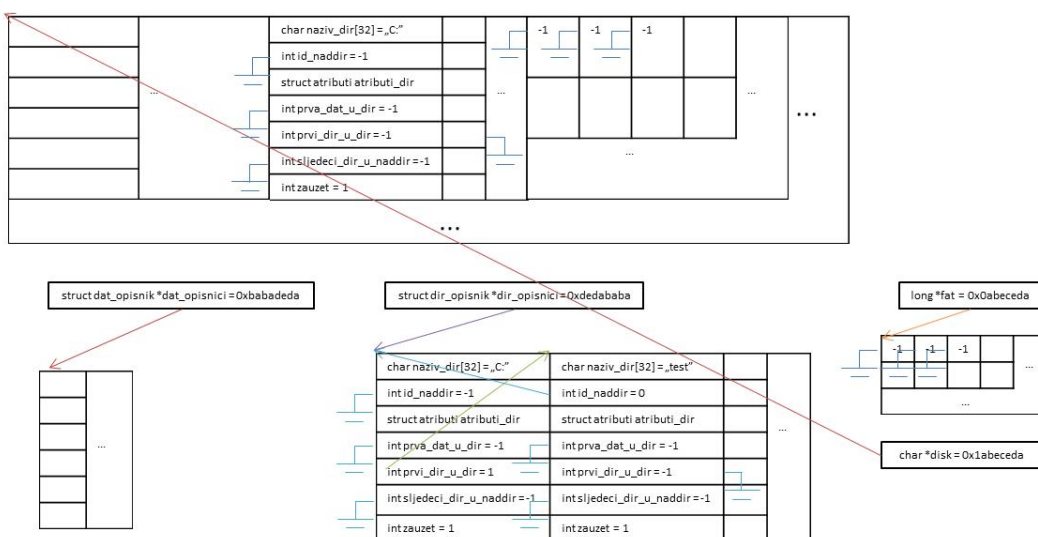
5.2. upali();



Slika 5.3: Stanje sustava nakon naredbe `upali()`;

Lijevo su datotečni opisnici. Odgovaraju prvomu sektoru na disku. Iznad njih je pokazivač. On pokazuje na njih. U sredini su direktorijski opisnici. Odgovaraju drugomu sektoru na disku. Iznad njih je pokazivač. On pokazuje na njih. Desno je FAT. Odgovara trećemu sektoru na disku. Iznad njega je pokazivač. On pokazuje na nj.

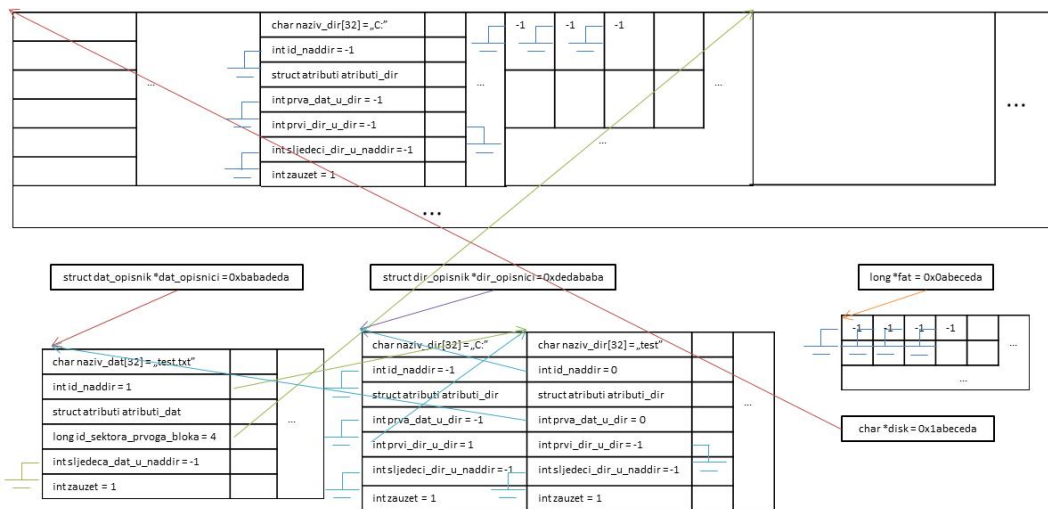
5.3. stvori_dir(„C:\\test“);



Slika 5.4: Stanje sustava nakon naredbe `stvori_dir(„C:\\test“)`;

Prikazana su samo tri direktorijska opisnika. Točkice pokazuju da ih ima još. Drugi je direktorijski opisnik popunjen. Njegov je naziv „test“. Njegov je identifikator naddirektorija 0. To znači da je njegov naddirektorij „C:“. Njegovi su atributi popunjeni. Njegova je prva datoteka u direktoriju -1. To znači da nema datoteka u njemu. Njegov je prvi direktorij u direktoriju -1. To znači da nema direktorija u njemu. Njegov je sljedeći direktorij u naddirektoriju -1. To znači da je zadnji direktorij u naddirektoriju. Njegova je zauzetost 1. Prvi direktorij u direktoriju prvoga datotečna opisnika jest 1. To znači da je njegov prvi direktorij „test“.

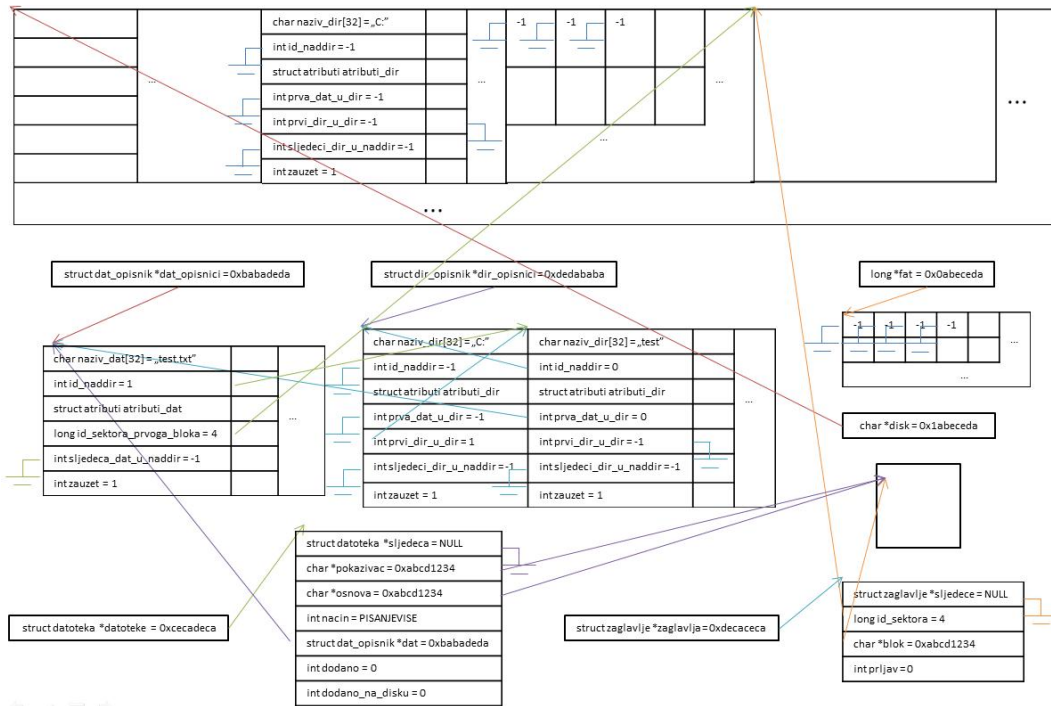
5.4. stvori_dat(„C:\\test\\dat.txt“);



Slika 5.5: Stanje sustava nakon naredbe stvori_dat(„C:\\test\\dat.txt“);

Prikazana su samo četiri sektora. Točkice pokazuju da ih ima još. Prikazana su samo dva datotečna opisnika. Točkice pokazuju da ih ima još. Prvi je datotečni opisnik popunjen. Njegov je naziv „test.txt“. Njegov je identifikator naddirektorija 1. To znači da je njegov naddirektorij „test“. Njegovi su atributi popunjeni. Njegov je identifikator sektora prvoga bloka 3. To znači da je njegov prvi sektor ovaj novi. Njegova je sljedeća datoteka u naddirektoriju -1. To znači da je zadnja datoteka u direktoriju. Njegova je zauzetost 1. Prva datoteka u direktoriju drugoga direktorijskoga opisnika je 0. To znači da je njegova prva datoteka „dat.txt“. Prikazano je samo deset zapisa u FAT-u. Točkice pokazuju da ih ima još. Četvrti je zapis -1. To znači da je zauzet.

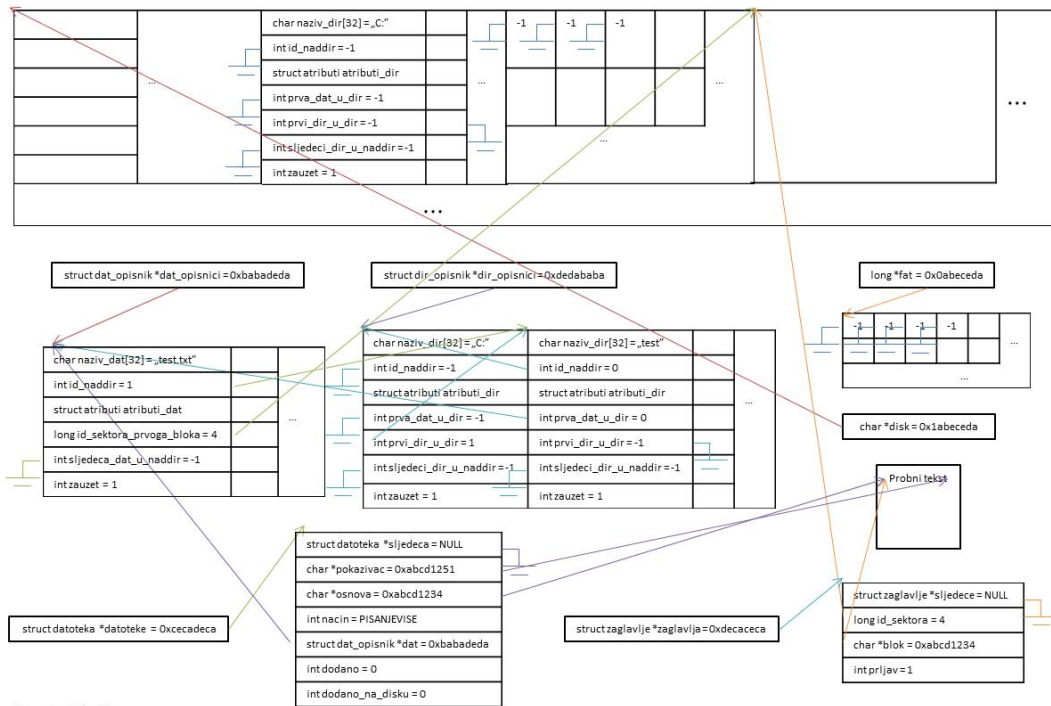
5.5. otvori_dat(„C:\\test\\dat.txt“, „r+b“);



Slika 5.6: Stanje sustava nakon naredbe `otvori_dat(„C:\\test\\dat.txt“, „r+b“);`

Ovaj je kvadrat novi blok memorije. Dolje u sredini lijevo jedan je opisnik otvorene datoteke. Popunjen je. Njegov je pokazivač sljedeće datoteke uzemljen. Njegov trenutni pokazivač pokazuje na blok memorije. Njegov osnovni pokazivač također. Njegov je način otvaranja datoteke `r+b`. Njegov pokazivač datotečnoga opisnika pokazuje na datoteku `„dat.txt“`. Njegov je brojač dodanih bajtova 0. Njegov je brojač dodanih bajtova na disku 0. Pokazivač je lijevo od njega. On pokazuje na nj. Dolje desno desno jedan je opisnik zaglavlja bloka otvorene datoteke. Njegov je pokazivač sljedećega zaglavlja uzemljen. Njegov je identifikator sektora 3. To znači da je njegov odgovarajući sektor onaj novi. Njegov blok pokazuje na novi blok. Njegova je prljavost 0. Pokazivač je lijevo od njega. On pokazuje na nj.

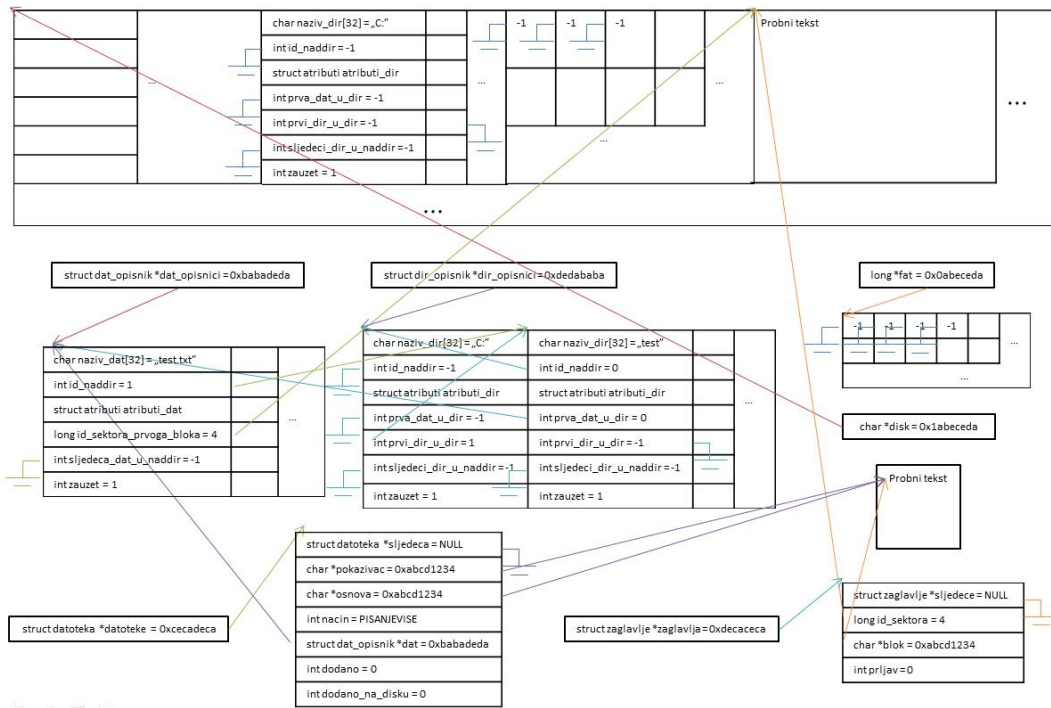
5.6. pisi_u_dat(„Probni tekst“, 1, 13, tok);



Slika 5.7: Stanje sustava nakon naredbe `pisi_u_dat(„Probni tekst“, 1, 13, tok)`;

U prvome bloku piše „Probni tekst“. Trenutačni pokazivač prvoga opisnika otvorene datoteke pokazuje iza ništičnoga znaka (`\0`). Prljavost prvoga opisnika zaglavlja bloka otvorene datoteke jest 1.

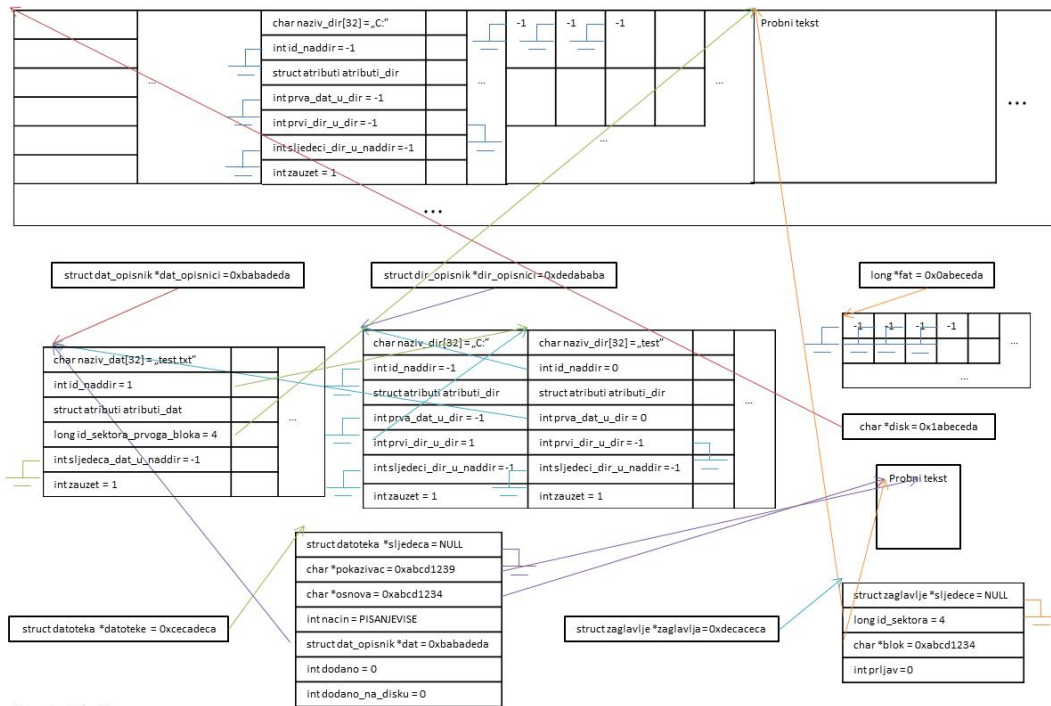
5.7. ponovno_otvori_dat(tok);



Slika 5.8: Stanje sustava nakon naredbe `ponovno_otvori_dat(tok)`;

U novome sektoru piše „Probni tekst“. Osnovni pokazivač prvoga opisnika otvorene datoteke pokazuje na novi blok. Prljavost prvoga opisnika zaglavlja bloka otvorene datoteke jest 0.

5.8. citaj_iz_dat(medusprennik, 1, 5, tok);



Slika 5.9: Stanje sustava nakon naredbe `citaj_iz_dat(medusprennik, 1, 5, tok)`;

Trenutačni pokazivač prvoga opisnika otvorene datoteke pokazuje na razmakni znak (x20).

6. Zaključak

U ovome je projektu napravljen datotečni sustav za jednostavan operacijski sustav oszur2010/k5. Sučelje sadržava sve bitne operacije. Implementacija sadržava sve bitne strukture podataka. Ostvareno je i čitanje određenoga sektora s diska i pisanje određenoga sektora na disk kroz komunikaciju s kontrolerom.

Ovaj je datotečni sustav koristan samo u jednostavnim operacijskim sustavima jer ne podržava mnogo naprednih funkcija i jer je komunikacija s diskom napravljena jako naivno. Ipak se ne može odbaciti kao neupotrebljiv jer podržava mnoge stvari, poput rekurzivnoga kopiranja direktorija, i jer radi s najnovijom današnjom specifikacijom komunikacije s diskom, AHCI 1.3. Zapravo, mogao bi se potpuno uključiti u bilo koji operacijski sustav. Njegovo se uspješno uključenje u sustav oszur2010/k5 može vidjeti u izvornome kodu.

LITERATURA

James Boyd. *Serial ATA Advanced Host Controller Interface (AHCI)*. Intel Corporation, MS: JF2-53, 2111 NE 25th Avenue, Hillsboro, OR 97124, 1.3 izdanju, lipanj 2008.

N/A. *Serial ATA*. 3855 SW 153rd Drive, Beaverton, Oregon 97006 USA, 3.0 izdanju, lipanj 2009.

Curtis E. Stevens. *Information technology - AT Attachment 8 - ATA/ATAPI Command Set (ATA8-ACS)*. Western Digital Technologies, 20511 Lake Forest Dr., Lake Forest, Ca. 9230, USA, 4a izdanju, svibanj 2007.

Datotečni sustav

Sažetak

U ovome se projektu opisuje ostvarenje jednostavnoga datotečnoga sustava, kao i osnovna komunikacija s kontrolerom AHCI. Najprije se opisuje sučelje AHCI, zatim upotreba sučelja, onda datotečni sustav, pa primjer rada.

Ključne riječi: datotečni sustav, *hard disk*, HBA, SATA, AHCI, ATA, PCI, DMA

File system

Abstract

In this project the realization of a simple file system is described, and also the basic communication with an AHCI controller. At the beginning the AHCI interface is described, then the use of the interface, then the file system, at the end an example of operation.

Keywords: file system, hard disk, HBA, SATA, AHCI, ATA, PCI, DMA