

(54) МЕТОД ЗА ДОСТЪП В КОМПЮТЪРНА СИСТЕМА И КОМПЮТЪРНА СИСТЕМА**Област на техниката**

Изобретението се отнася до метод за достъп в компютърна система и компютърна система, с които се осигуряват дублиращи копия от данни, запаметени в запомнящо устройство, за да се предпази системата от възможността запомнящото устройство да греша или данните да бъдат разрушени, загубени или заразени с компютърен вирус. Това изобретение се отнася по-специално до осигуряване на дублиращи копия за персонални компютри.

Обикновено в простите компютърни системи за един ползвател запамятащото средство от вида на хард- или фиксиран диск само една приложна програма има достъп до запамятащото средство в който и да е момент. Задачата за дублиране на данни, запаметени от запамятащо средство, е предадена чрез приложна програма с изключителна употреба на запамятащото средство, докато се провежда дублиращата операция, и така в подобна среда никоя друга приложна програма не може да се изпълнява, докато не завърши дублиращата операция.

С въвеждането на многоцелевите среди, които позволяват да се извършват едновременно няколко задачи, и на мрежовите системи, които позволяват на няколко ползвателя да ползват едновременно едни и същи източници, може да възникне ситуация, при която повече от едно приложения имат конкурентен до едно и също запамятащо средство. Когато такава ситуация възникне, има възможност за конфликт между заявките. Например, едно искане може да модифицира или да изтрие файл, който е бил използван от друго искане. Това довежда до смущения и потенциално до възможност за разрушаване на запамятащото средство и съответно до загуба на данни.

Известните методи "заклучване на файл" и "заклучване на запис" са въведени в операционните системи, за да се избегнат, в случай на заключване на файл повече от едно искане да се запише във файла, а в случай на заключване на запис, две иска-

ния, модифициращи същата област на даден файл. Тези методи са достатъчно задоволителни за повечето заявки, по-точно, ако всяко искане прави само незначителни промени във файловете, които се употребяват едновременно, и поради това не внасят значително закъснение за другите заявки, желаещи достъп до файла или записа, който е заключен. Специален е случаят на заявка, дублираща периодично данните в компютърната система, в която запамятащото средство се използва едновременно.

Най-малката единица от данни, която може да се дублира, е единичен файл, защото нито дублиращото искане, нито операционната система познават вътрешната структура на файла, тъй като той би могъл да бъде използван съвместно с всяка приложна програма.

Освен това, дублиращото копие на файла се нуждае от това да бъде точно изображение на оригинала в определен момент от време. Копирането на файла, все пак, не е мигновен процес и времето, необходимо за копирането му, зависи от размера на файла и от скоростта на копиращата операция. Това води до проблем при дублиращото искане - да се осигури състояние, при което никакво друго искане да не изменя файла, докато трае процесът на копиране. За да се копира даден файл, дублиращото приложение определя една област от памет със случаен достъп (РАМ) като пренасящ буфер и обработва чрез последователно четене блокове от източника на файла в буфера и след това записва съдържанието на буфера във второ запамятащо средство. Даже и файлът да е много малък, предаваният буфер се запълва няколко пъти и операцията по копирането изисква няколко операции четене и запис до пълното изкопиране на файла.

Потенциалният проблем, предизвикан от разрешението файлът да бъде изменян в процеса на копиране, може да бъде илюстриран от разглеждането на примера на файл, който е бил копиран като последователност от 10 блока. Ако файлът е бил изменен от транзакция, която изисква изменения на файла, кореспондиращ с блокове 3 и 8, след като блоковете от 1 до 5 вече са били изкопирани, копиращият процес впоследствие би изкопирал измененията в блок 8, но няма да

изкопира измененията в блок 3, тъй като този блок вече е бил изкопиран, преди транзакцията да е била извършена. Дублиращото копие съдържа само частично транзакцията, което може да направи дублиращото копие напълно ненужно, тъй като искането, което го "признава", вероятно ще прецени, че файлът трябва да бъде разрушен.

В патент US 5 086 502 се разкрива метод за дублиране на всяка операция запис, която се извършва чрез дублиран запис на операцията във вторично запамятаващо средство. В добавка към този метод, има два други начина за дублиране на файлове в компютърна система, имаща едновременно използвано запамятаващо средство:

а) заключване на файла, докато се дублира. Това предпазва другите ползватели от изменение на файла чрез отказ на достъп за четене. Ако друг ползвател има вече достъп за четене, дублиращото искане трябва или да пропуска този файл, или да чака, докато може да се получи достъп с изключителен запис.

б) откриване на измененията във файла. След копиране на файла, дублиращото искане може да използва операционната система да провери дали по време на копиращата операция е извършена някаква промяна. Ако промяна е била извършена, копирането трябва да се повтори или този файл да се пропусне.

Нито един от последните два метода не осигурява едно адекватно решение, защото много файлове (основно бази данни) са постоянно в употреба. Нещо повече, двата метода изискват ползвателите да бъдат защитени ефективно, докато се извършва дублирането. Допълнително, малко вторични запамятаващи средства и комбинации операционни системи са в състояние да осъществят скорост на предаване на данни, повисока от 15 мегабайта за минута, и поради това времето, необходимо за да се изкопира голям файл, е значително.

Няколко приложни програми преодоляват проблема чрез осигуряване на методи за сигурно дублиране на собствените файлове в средства за едновременно ползване, но процедурата за всеки от тях е различна. За дублиране на цялото запамятаващо средство дублиращото приложение изисква да се

знае как работи всяка различна вградена процедура или трябва да се опира на ръчно въздействие в една определена точка.

Поради това, че дублиращото искане трябва да дублира целия файл и че той трябва да не бъде променен от друга приложна програма, докато трае процесът на дублиране, използването на известните до момента методи въвеждат закъснения в компютърната система, имаща запамятаващо средство за едновременен достъп. Това частично е вярно при дублиране на големи файлове.

В патент на US 5 163 148 се описва дублираща система, в която, преди да се стартира дублирането на даден файл, той е доведен до устойчиво състояние, например, ако са заявени множество операции във връзка с транзакции, тогава, когато всички тези операции са приключени. Ако една част от файла е била дублирана и изисква актуализиране, тогава тази част от оригиналния файл е изкопирана в отделна част на диска, преди да е извършена актуализацията. Програмата за дублиране дублира файла чрез замяна на тези части, които са били актуализирани, чрез оригиналните части, които са били изкопирани, и ги запамятава някъде върху диска. Непрекъснатата област на диска с пряк достъп, резервирана за данни за съответствие, която обикновено съдържа списък на пътеките и секторите на диска, заети от този файл, е разширена и съдържа азбучен указател, свързан с разположенията на пътеките и секторите на откопираните и на оригиналните части на файла до разположенията на пътеките и секторите на актуализираните части, които заемат области, от които те са били откопирани.

В патент на WO95/19599 се описва метод за работа на компютър, имащ едно запамятаващо устройство за едновременен достъп от няколко приложни програми, който, също така, обхваща дублиращ процес, състоящ се от процес за осъществяване на запис на измененията, направени в данните на запамятаващото устройство, докато дублиращият процес не е завършен. Записът е използван, за да създаде сигурност, че данните, проверявани от дублиращия процес, не са променени, докато тази операция се е извършвала, като се допуска други процеси да продължат нормално да дублират запамята-

ващото устройство, при това - избягвайки закъсненията, съпровождащи техниките за заключване на файла. Допълнителните операции са изпълнени на нивото на интерфейса на операционната система на микрокомпютър по същия начин, така че нормалното използване на компютъра не е засегнато. В този метод дублиращият процес е в състояние да генерира един старт и един стоп сигнал и да изпълни най-малко една дублираща операция четене, за да прочете оригиналните данни от запаметяващото средство, например, от харддиска. Централното обработващо устройство изпълнява най-малко една нормална операция запис, за да запаметява периодично в първа област на запаметяващото средство, както и най-малко една нормална операция четене, за да прочете данни от запаметяващото средство.

След получаване на споменатия старт сигнал, се изпълнява една операция на прекъсване, за да се прекъсне операцията запис и да се запишат данни във втора област на запаметяващото средство, така че да се предпазят оригиналните данни, върху които биха могли в противен случай да се запишат данните от операцията запис. Записът на разположението на файла, направен във втората област, е поддържан и при изпълнение на дублираща и нормална операция четене е осъществен достъп до записа, за да се прочетат съгласно записа данните от една от двете - първата или втората, области на запаметяващото средство. По този начин, процесът на дублиране е осигурен със запазване на оригиналните данни в състоянието, в което са били непосредствено преди генерирането на старт сигнала. В този миг записът дефинира частите на файла в условия на запис на файла, но не в условия на пътека и сектор на разположение върху диска. Това избягва проблемите, които могат да възникнат, ако записът е направен в условия на разположение върху диска, т.е. пътека и сектор, и осигурява определени предимства.

За предпочитане записът обхваща една област делта (с три свързани части), в която са запаметени първите данни и едно указание за преобразуване между действителното място на първите данни и неговото очаквано разположение. Алтернативно, вторите данни могат да бъдат запаметени в ед-

на такава област, а другите операции могат да продължат за се записват на очакваното място в запаметяващото средство.

За предпочитане делта областта е различна област на същото запаметяващо устройство, т.е. на харддиска. Заявените допълнителни операции предимно са запаметени и осъществени на нивото на интерфейса на операционната система на компютърната система. Ако софтуерът е запаметен на ниво операционна система, той може да бъде направен прозрачен за ползвателите, така че да не се свързва към приложния софтуер.

Това дублиращо записване може да бъде запаметено в различни запаметяващи средства, например лентово запаметяващо устройство, оптичен диск или в друга област на основната запаметяваща среда. Щом дублирането веднъж е извършено, то може да бъде възстановено по конвенционален начин.

Техническа същност на изобретението

Методът за достъп в компютърна система, базираща се на микропроцесор и съдържаща памет за инструкции, централно управляващо устройство и запаметяващо средство, включва следните операции:

- осъществява се дублиращ процес, извършван от споменатата компютърна система, която генерира старт сигнал и стоп сигнал и най-малко една дублираща инструкция за четене на налични данни от споменатото запаметяващо средство; като

- се извеждат (четат) първа група инструкции, запаметени в споменатата памет, към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запаметяващото средство, а споменатият дублиращ процес се осъществява с файлове, които са отворени за достъп от споменатите инструкции; при което

- се поддържа записът на данните, записани в запаметяващото средство, и се свързват съответстващи части на първа запаметяваща област на споменатото запаметяващо средство и втора запаметяваща област на същото запаметяващо средство; като

- след получаване на споменатия старт сигнал и получаване на инструкция за че-

тене се изпраща запитване до споменатия запис за вида на операцията за четене - за извършване на дублираща или нормална операция за четене, след което се четат данни от една от споменатите първа и втора запамятаващи области към този запис и същевременно се изпълнява и другата от споменатите дублираща и нормална операция за четене, четейки данни от споменатата първа запамятаваща област, и

- се извършва и операция за проверка на областта от паметта, от която произлиза инструкцията за четене и се идентифицира програмата, която е инициирала споменатата операция за четене.

Освен това, методът за достъп в компютърна система, базираща се на микропроцесор и работеща в компютърна мрежа, осъществява и операция, при която се присвояват (записват) към споменатия дублиращ процес идентификационни данни за ползвател, процес и/или задача и се проверява за съответствие на споменатата инструкция за четене към дублиращата операция за четене и/или нормална операция за четене, отговаряща на заявената от ползвателя и/или по програмен път (по мрежата) със споменатите идентификационни данни за ползвател, процес и/или задача.

Операцията за проверка на областта на паметта се осъществява чрез запитване на стека и се идентифицират местата със запаметени данни в стека и местоположението им.

Същата операция за проверка на областта на паметта представлява сравняване на кода на споменатите местоположения с кода, предвиден от дублиращата програма.

Тази операция за проверка на областта от паметта представлява и идентифициране на Program Segment Prefix и на Environment String, асоциирани към програмата в споменатите места, като същевременно се идентифицира името на програмата, използваща Environment String.

Възможно е при метода за достъп в компютърна система, базираща се на микропроцесор и съдържаща памет за инструкции, централно управляващо устройство и запамятаващо средство, при което се осъществява дублиращ процес, извършван от компютърната система, която генерира

старт-сигнал и стоп-сигнал и най-малко една дублираща инструкция за четене на налични данни от споменатото запамятаващо средство; като се подават инструкции от паметта към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запамятаващото средство; и се използва средство за отваряне на файл, който се чете чрез нормална операция за четене, и същевременно средството забранява достъпа до споменатия файл за други операции за четене, да се блокира забраняващата функция на средството за отваряне на файл и да се разрешава достъп за четене до споменатия файл при едновременно осъществяване на дублираща операция за четене и нормална операция за четене.

Освен това компютърната система съхранява една таблица на отворените файлове и забраняващата функция се блокира чрез временно обратно изменение на таблицата. Генерирането на споменатия старт сигнал се забавя до момента, в който завърши който и да е последователност от операции за запис във файла, който трябва да бъде дублиран.

Компютърната система съдържа основно микропроцесор и памет за инструкции, централно управляващо устройство и запамятаващо средство, при което към структурата на компютърната система са предвидени и следните средства:

- средство за осъществяване на дублиращ процес, извършван от компютърната система, която генерира старт сигнал и стоп сигнал и най-малко една дублираща инструкция за четене на налични данни от запамятаващото средство;

- средство за извеждане (четене) на първа група инструкции, запаметени в паметта, към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запамятаващото средство, а дублиращият процес се осъществява с файлове, които са отворени за достъп от инструкциите;

- средство, което поддържа запис на данните, записани в запамятаващото средство, и свързва съответстващи части на първа запамятаваща област на запамятаващото средство и втора запамятаваща област на съ-

щото запаметяващо средство;

- средство, което след получаване на старт сигнал и получаване на инструкцията за четене изпраща запитване до споменатия запис за вида на операцията за четене - за извършване на дублираща или нормална операция за четене, след което чете данни от една от споменатите първа и втора запаметяващи области към този запис и същевременно изпълнява и другата от споменатите дублираща и нормална операция за четене, четейки данни от първата запаметяваща област, и

- средство, което извършва и операция за проверка на областта от паметта, от която произлиза инструкцията за четене, и идентифицира програмата, която е иницирала операцията за четене.

Компютърната система, базираща се на микропроцесор и работещата в компютърна мрежа, има

- средство за присвояване (записване) към споменатия дублиращ процес на идентификационни данни за ползвател, процес и/или задача и за проверка за съответствие на споменатата инструкция за четене към дублиращата операция за четене и/или нормална операция за четене, отговаряща на заявената от ползвателя и/или по програмен път (по мрежата) със споменатите идентификационни данни за ползвател, процес и/или задача.

В структурата на компютърната система е предвидено средство за осъществяване на операцията за проверка на областта на паметта чрез запитване на стека и идентифициране на местата със запаметени данни в стека и местоположението им, като средството за осъществяване на операцията за проверка на областта на паметта сравнява кода на споменатите местоположения с кода, предвиден от дублиращата програма.

Освен това предвиденото средство за осъществяване на операцията за проверка на областта от паметта идентифицира Program Segment Prefix и Environment String, асоциирани към програмата в споменатите места, като същевременно идентифицира името на програмата, използваща Environment String.

Компютърната система, базираща се на микропроцесор и съдържаща памет за инструкции, централно управляващо устройство

и запаметяващо средство, и която съдържа следните средства:

- средство за осъществяване на дублиращ процес, извършван от компютърната система, която генерира старт сигнал и стоп сигнал и най-малко една дублираща инструкция за четене на налични данни от запаметяващото средство;

- средство за подаване на инструкции от паметта към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запаметяващото средство; и

- средство за отваряне на файл, който се чете чрез нормална операция за четене, и същевременно средството забранява достъпа до споменатия файл за други операции за четене; има и

- средство, което блокира забраняващата функция на споменатото средство за отваряне на файл и разрешава достъп за четене до споменатия файл при едновременно осъществяване на дублираща операция за четене и нормална операция за четене.

Предвидено е средство за съхраняване на таблица на отворените файлове, което блокира забраняващата функция чрез временно обратно изменение на таблицата.

Средството за генериране на старт сигнала има структура, която забавя генерирането до момента, в който завърши която и да е последователност от операции за запис във файла, който трябва да бъде дублиран.

Пояснения на приложените фигури

Фигура 1 е блокова схема на персонален компютър и касетно лентово запаметяващо средство.

Фигури 2А и 2В са схеми, илюстриращи изображението на паметта на системата на нивото на операционната система, съответно в конвенционален персонален компютър и в компютър, модифициран да работи в съответствие с метода от изобретението.

Фигура 3 е блокова диаграма на метод за запис на данни при дублираща операция.

Фигура 4 е блокова диаграма на второ примерно изпълнение на метода от фигура 3.

Фигура 5 е блокова диаграма, илюстрираща едно усъвършенстване на примерните изпълнения от фигури 3 и 4.

Примери за изпълнение на изобретението

Съгласно фигура 1 персоналният компютър 10 обикновено съдържа централно обработващо устройство 12, памет с произволен достъп 14 и запамятаващо устройство, способно да не загубва информацията при прекъсване на захранването в изпълнение на управление на харддиска 16. В този пример за целите на осигуряване на дублиращо запамятаващо средство към компютъра 10 е свързано лентово устройство 18.

Част от дублиращата програма е оформена като записващ процес, който прекъсва нормалните операции четене и запис от и към управлението на диска 16, с цел да контролира потока данни между управлението на диска 16 и останалите части на компютъра 10.

Преди стартиране на процеса за изготвяне на дублиращо копие от даден файл, запамятаващият процес е активиран чрез дублиращо приложение, за да пренасочи цялата активност на записа на диска към друга област от диска 16, наречена "делта" област. За да се осигури запис на операциите запис върху целия диск, които са били пренасочени към делта областта, се поддържа таблица за делта преобразуване. Тази таблица поддържа преобразуване между очакваните направления върху диска 16 на данни, асоциирани с всяка операция запис върху диска, от една страна, и тяхното действително място на запамятаване в делта областта, от друга страна. Ако следващата операция запис се случи в област, която вече е била обект на въвеждане в таблицата, старите данни в делта областта са разрешени за презаписване върху тях на новите данни. По този начин данните в главната част на диска не се променят, докато се извършва дублиране, тъй като всички промени са пренасочени към делта областта.

Когато е зададена една операция запис чрез дублиращото приложение, тя се пропуска непроменена, така че данните, върнати към дублиращото приложение, идват от оригинална област данни на главната част на диска. Когато е зададена операция четене

на диска чрез друго приложение, запамятаващият процес първо извършва проверка на таблицата, за да види дали има запис на диска в тази област, тъй като процесът стартира запитваща операция четене, и ако е така, операцията запис е пренасочена към кореспондиращо място в делта областта. Ако не се открие запис, ако тази част от данни не е била вече променена от стартирането на запитващата операция, тогава операцията четене се пропуска непроменена.

По този начин промените на данни, запаметени на диска 16, са преглеждани нормално от всички приложения, освен от дублиращото приложение. Дублиращото приложение преглежда данни, които няма да се променят, докато се генерира инструкцията да се спре записващият процес, изключващ операциите четене и запис.

Когато операцията за дублиращото копиране е завършена, записващият процес е активиран, за да върне компютъра към нормалното му опериране. Тази фаза на възстановяване е извършена чрез прилагане на измененията, запомнени в делта област към съответните области на оригинални данни в главната част на диска 16, използвайки преобразуванията, запаметени в таблицата. Докато делта измененията са били приложени, операциите запис на диска продължават да бъдат прекъснати, както беше обяснено по-горе. Когато няма повече операции запис, които са били приложени към оригиналните данни, прекъсването на операциите запис е отменено и системата оперира нормално.

При едно по-нататъшно усъвършенстване, ако един запис е съхранен на "екстента", към който процесът на дублиращо копиране е програмиран, софтуерът може да разреши операциите запис в главната част на диска 16 да бъдат пропуснати непроменени, след като тази част от диска е била дублирана, тъй като вече не е от значение дали данните се променят. Това намалява размера на таблицата за делта преобразуване. Също така, ако областта от главната част на диска, към която тази операция е насочена, е била вече обект на въвеждане в таблицата за преобразуване, това въвеждане може просто да бъде изтрито, при което се намалява времето, необходимо за фазата на възстановяване.

Изобретението е приложимо специално към IBM съвместими персонални компютри, което означава, към по-голямата част от системите, базирани на персонални компютри, използвайки като тяхно централно обработващо устройство процесорен чип от следните типови номера 8086, 8088, 80286, 80386, 80486 и Pentium, произведен от Intel, както и подобни микропроцесори. В такива компютри, когато програмата изисква да се осъществи достъп до файл, тя извиква стандартна програма, която записва данни върху диска. Тази програма, която в случая на DOS операционна система е известна като Interrupt 21 hex (INT 21h), е интегрална част от операционната система. Действията, изпълнявани от програмата, известна като стандартна програма на оператора на прекъсване, зависят от параметрите, пропуснати до програмата през входа. Тази програма е показана на фигура 2А като оформяща част на операционната система в един списък на разпределение на параметрите на системата, а точката на въвеждане на INT 21h е показана чрез стрелка. За провеждане на предпочитания метод съгласно изобретението е добавен допълнителен код на програмата на ниво интерфейс на операционната система, както е показано на фигура 2В. На практика в DOS среда, това може да бъде заредено в компютъра като драйвър на устройството използва файла CONFIG.SYS.

Допълнителният софтуер има ефекта на инструкция за запаметяване на данните, които са били пренесени от един алтернативен комплект от инструкции. Подобна техника може да бъде използвана, за да се прекъсне инструкцията за запис на данни от диска и същите да се пренесат чрез един алтернативен комплект от инструкции.

Съответна последователност от операции, осигурена от алтернативния комплект инструкции, е показан на блоковата диаграма на фигура 3. При получаване на повикване INT 21h, програмата първо проверява дали повикването е повикване запис. Ако е така, програмата проверява в стъпка 20 дали това предполага да бъде прекъснато записването на повикването. Ако не е, данните са записани (стъпка 22) в областта на оригиналните данни.

Ако исканията за запис са били пре-

късвани, тогава програмата проверява таблицата за делта преобразувания (стъпка 26), за да установи дали преди това в областта на оригиналния файл 24 е бил правен запис, определен в условията на специфицирани записи или на части във файла. Ако (стъпка 28) е бил правен предварително запис, тогава тази част от делта областта 30, предварително преобразувана в тази част от оригиналния файл 24 в таблицата за делта преобразуване, е презаписана с новите данни (стъпка 32).

Ако повикванията за запис са били прекъсвани, но предишен запис не е бил правен в тази част на оригиналния файл 24, тогава записът е добавен към таблицата и една нова област в делта областта е отредена на тази част от данните (стъпка 34). След това данните са записани в нова делта област (стъпка 36).

Ако повикването е повикване за четене, програмата проверява в стъпка 38 дали е очаквано да бъдат прекъснати исканията за четене. Ако не, тогава данните са четени от оригиналния файл 24 (стъпка 40).

Ако програмата е прекъсваща повикванията четене, тогава тя първо проверява в стъпка 42 дали повикването идва от приложна програма за дублиране. Ако повикването идва от дублиращо приложение, програмата преминава към стъпка 40 и данните са прочетени от оригиналния файл 24.

Ако достъпът за четене не идва от дублиращо искане, програмата проверява таблицата за делта преобразуване, за да провери дали запис в областта на оригиналния файл 24, кореспондиращ с този на повикването четене, е бил пренасочен към делта областта (стъпка 44). Ако не е бил пренасочен (стъпка 46), програмата пропуска обратно към стъпка 40 и данните са прочетени от оригиналния файл 24. Ако четенето е било пренасочено, данните са прочетени от правилната част на делта областта 30 (стъпка 48).

Ако повикването INT 21h не е нито повикване четене, нито повикване запис, то е върнато към нормалната стандартната INT 21h програма за управление.

Във второто примерно изпълнение на изобретението, когато операциите четене и запис са били прекъсвани, докато е извър-

шван процеса на запис, вместо запаметяване на променените данни в делта областта, когато първият запис в определена област на файла е направен, оригиналните данни са запаметени в делта областта и на оригиналния файл е разрешено да се променя. Направено е едно въвеждане в таблицата за преобразуване, което показва оригиналното съдържание в тази област на оригиналния файл, който сега е запаметен в делта областта. Всеки следващ запис в същата област на файла е игнориран чрез процеса на записване. Операциите четене от дублиращото приложение след това са пренасочени към делта областта, ако таблицата на преобразуване показва, че тази област от оригиналния файл е била презаписана.

На фигура 4, второто примерно изпълнение е показано чрез блокова диаграма. Диаграмата е подобна на тази от фигура 3 със стъпки, имащи същите номера, означаващи подобни действия.

Разликата между двете фигури (3 и 4) са следните.

В стъпка 60, ако е активирано прекъсване на четене (стъпка 38) и четенето не е от дублиращо приложение, данните са прочетени от оригиналната област данни. Това е включване на обратен ход от стъпка 42 на фигура 3.

Една допълнителна стъпка 62 е въведена след стъпка 28, която чете данни от оригиналната област за данни, преди да е презаписана за първи път.

Стъпка 36 от фигура 3 е модифицирана, за да оформи нова стъпка 64, която инкорпорира един допълнителен запис в делта областта на данни, прочетени в стъпка 62.

Стъпка 32 е модифицирана, за да оформи една нова стъпка 66, която не пренасочва данните за презапис в делта областта, а вместо това записва данните върху областта на оригиналните данни.

Този втори метод е по-бавен от първия, защото всеки достъп за запис във файла се преобразува в едно четене и два записа, за да се предотврати единичният пренасочен запис на първия метод. Вторият метод не изисква измененията, запомнени в делта областта, да се прилагат към оригиналния файл щом дублирането е завършено. Таблицата за делта преобразуване и данните, за-

паметени в делта областта, могат просто да бъдат освободени. Освен това, поради тази разлика, не е необходимо да бъде направено осигуряване за предпазване от фатални ситуации (таква като прекъсване на храненето), възникващи, когато делта измененията не са били напълно приложени към оригиналните данни.

В случай че описаният по-горе метод изпълнява част от дублиращото искане, прекъсването на операция четене може да се извърши в рамките на дублиращото приложение и не се нуждае от ангажиране употребата на програмата на операционната система.

Детайлната реализация на системата, както е описана във връзка с фигури 3 и 4, е в компетенциите на средния специалист в областта, и поради това не е необходимо по-подробно описание. Ако се реализира в MS.DOS, за четящото устройство се отнесе- те, ако е необходимо, към IBM DOS Technical Reference Manual като основно помагало.

Изобретението е реализирано, за предпочитане, като част от самата операционна система или чрез въвеждането ѝ като неразделна част на кода на оригиналната операционна система, или чрез формиране на нейно разширение, във формата на драйвер на устройството. В случай на компютър Intel 80386 (и по-висок) изобретението би се реализирало типично на най-високото привилегировано ниво, известно като "Ring 0", като достъпът до всички елементи в системата е неограничен.

Изобретението не се опира на някаква определена архитектура на компютъра при реализирането му. То може да бъде реализирано лесно на други системи, например, на компютри "Макинтош", които използват серии 6800 и Power PC.

Изобретението може да бъде използвано в среда на много потребители. В този случай е препоръчително изобретението да бъде използвано в среда с контрол на достъпа до файловете. Изобретението ще осигури неразрушаващо дублиращо копие, използвано, докато останалите части на системата работят ефективно точно така, както биха работили, ако изобретението не съществува. При липса на контрол на достъпа до файловете има опасност от разрушаване на фай-

ла, независимо дали изобретението е използвано или не.

По-специално изобретението е приложимо към файлове, които може да бъдат големи и да са разделени в записи. Записите, които са актуализирани, са записани в таблица и двете - актуализираната и оригиналната версия на записа, са поддържани, докато дублирането е завършено. Таблицата може да съдържа, например началната точка във файла, в която започва актуализирането на данните, и дължината, кореспондираща с количеството на данните, които са актуализирани. Алтернативно, таблицата може да съдържа и условията на идентичност на актуализираните записи.

При ситуацията на мрежа от много потребители, когато компютърната среда позволява идентифицирането на определен ползвател, който иска разрешение да изпълни дублиране, методът може да съдържа и стъпки на разпознаване на ползвателя и дублиране на файла само за този ползвател по начина, който беше описан по-горе. По този начин не е необходимо да се блокират всички транзакции на данни, а само на тези, които са свързани с този конкретен ползвател.

При идентификация на източника на повикване методът за дублиране, описан по-горе, който може да оперира с файлове, които са отворени, се основава на възможността да се различат повикванията четене от дублиращото приложение и тези от други източници, например от нормални приложения. Това възниква на стъпка 42 от фигура 3 или на стъпка 60 от фигура 4.

Ако методът на отворения файл е осигурен като интегрална част на дублиращото приложение, източникът на повиквания четене може да бъде определен лесно, защото дублиращото приложение "знае" кои запитвания генерира то. Проблем възниква, ако дублиращото приложение вече присъства, например във формата на наличен стандартен пакет или в друг случай, във форма, в която не може да бъде изменено, за да акомодира метода на отворения файл, който трябва вместо това да бъде реализиран чрез допълнителен код.

В съответствие с първото усъвършенстване запитванията от дублиращото прило-

жение са различни от останалите запитвания чрез проследяване на веригата събития (като преглеждане назад или "възходящо", за да се идентифицира областта на кода, който е иницирал повикването четене. Следващите примери за използване на този метод в MS.DOS операционна система отново ще бъдат използвани като илюстрация, а подобна техника може да бъде приложена и към среди на други операционни системи.

Повикването четене е иницирано от INT 21h повикване, което нормално би било насочено директно към програмата на оператора за прекъсване. Програмата на оператора за прекъсване, обикновено няма представа от повикващия. Тя просто казва да се извърши определена операция и след това да се върне към повикващия. В компютрите Intel, споменати по-горе, които са използвани в DOS среди, компютърът автоматично поставя върнатия адрес на ползвателя на стека и прескача на първата инструкция от програмата за прекъсване. Когато функцията е била извършена, прекъсващият оператор генерира една инструкция IRET (връщане на прекъсването), която предизвиква компютъра да извлече обратния адрес от стека и да продължи изпълнението на обратния адрес. По този начин прекъсващият оператор се нуждае да има и наистина няма знание за повикващия.

В системите, описани по-горе съгласно фигури 3 и 4, прекъсващата програма INT 21h е инсталирана от един алтернативен комплект от инструменти. В допълнение на функциите, описани по-горе, тези инструкции може да бъдат адаптирани към допълнително определяне на източника на повикване, както следва.

В това усъвършенстване прекъсващият оператор оценява стека, за да определи обратния адрес. В случай на компютър Intel серия 80X86 кодът на отместване и сегментът на запитващия ще бъдат запаметени като първа и втора дума отгоре на стека, т.е. на SS:[SP] и SS:[SP+2], съответно. Едно описание на стъпките, извършвани при една MS.DOS реализация на това усъвършенстване, са дадени като приложения към това описание. Стандартната програма след това може да погледне източника на повикването,

както е дефиниран от стека, и да определи програмата, която е направила повикването по множество от възможни начини, два примера от които са описани по-долу:

а) Тъй като кодът във всяка версия на наличната програма е един и същ, различните характеристики на кода относно обратния адрес могат да бъдат изследвани, за да се определи дали повикващият е известна дублираща програма. Кодът на обратния адрес е сравнен с кода, очакван от дублиращата програма, която е била използвана, за да се определи, дали има подходяща индикация, че запитването е било направено от дублиращата програма.

б) Алтернативно името на програмата може да бъде установено чрез отнасяне до Program Segment Prefix (PSP) и Environment String, заедно с всяка заредена програма. От стека сегментът е известен и PSP може да бъде прочетена. Това ще разреши да бъде определен "низът" на средата, а следователно и името на програмата. Тази техника е описана пълно в справочника "Undocumented DOS" от Andrew Schulman и други, публикувани от издателство Addison-Wesley, второ издание 1993, ISBN 0-201-63287-X, глава 7, стр. от 356 до 359.

В случай на работа на системи, различни от MS.DOS, идентичността на повикващия може да бъде определена от други параметри на стека, съдържащ номера за връзка, номера за идентичност на задачите, номера на процеса и други подобни. В определена операционна система също могат да се използват глобални вариации, за да се определи името на процеса.

Методът се основава на идентифициране на някои известни характеристики на повикващия, такива като име на програмата, име на процеса или чрез идентифициране на някои "сигнатури" в кода на програмата, които могат да бъдат известна поредица от байтове, чрез преглеждане през стека обратно на програмата за повикване.

В съответствие с второто усъвършенстване, което може да бъде използвано в среда на мрежов сървър, идентичността на програмата за повикване е определена от резервиране на определено име на ползувател само за целите на дублирането. Вместо установяване на идентичността на програмата,

името на ползувателя (уникална идентификация на ползувателя) е идентифицирано, и всички повиквания от резервирания ползувател са присвоени на кода от дублиращата програма. Тъй като всеки ползувател е по-молен да въведе името си в сървъра преди достъпа до файловете да е разрешен, това е един ефективен метод, осигуряващ, разбира се, че това име на ползувателя е използвано винаги за дублиране, а не за някаква друга цел. Този метод не изисква предварително познаване на определена дублираща програма.

По подобен начин идентификация на процеса или на задачата, специфична за дублиращия процес, може да бъде изискана, за да се определи дали тя е дублиращ процес или друго искане, което е предизвикало четене. Две задачи с една и съща идентификация (id) никога не присъстват в едно и също време, така че те идентифицират процеса уникално във всеки миг (пропусканите номера на задачата са използвани отново).

Описаното по-горе се съдържа и в искане към заявка за четене, но то може да бъде използвано чрез прилагане на същата процедура към повикване запис. В случай че тази дублираща програма генерира заявка за достъп за запис, тогава може да се извършат определени действия, които нормално ангажират някоя грешна програма, тъй като при дублиращото искане нормално не би трябвало да бъде генериран достъп за запис към прекъсващия оператор.

Както е описано по-горе, допълнителните инструкции първо определят дали прекъсващото повикване е повикване четене (или запис), и, ако е така, определят дали запитването е генерирано от дублиращата програма или от друго приложение. Принципно, редът на тези две операции би могъл да бъде променен и първо да бъде определен източникът на повикване, а после да се определи естеството на повикването, което се проучва.

Методът за дублиране на отворен файл, описан с помощта на фигура 3 и 4 предполага, така че достъпът за четене на файлове винаги може да бъде получен чрез дублиращо искане. На практика това не винаги е така, тъй като файлът може да е бил отворен за изключителна четене/запис употреба

от друга заявка и по този начин достъпът може да бъде отказан за дублиращото искане.

Съгласно изобретението се оценява, че, тъй като дублиращото искане никога не би трябвало да записва във файла, то е дублиране, и поради това, че описаният по-горе метод на отворения файл ще осигури копие-то на файла, получено чрез дублиращо искане да се променя, докато се дублира, в действителност за дублиращо искане напълно е безопасно да има достъп за четене до файла.

Достъп за дублиращи цели може да бъде осигурен по следния начин. Операционната система обикновено съдържа една таблица на всички отворени файлове, която съдържа информация като идентификатори на оператора на файла, указатели на отстъпване на файла, права на достъп, поддържани за всеки отворен файл.

В случай на MS.DOS операционна система, файловете са отворени, използвайки INT 21h функцията. Така повикванията на отворените файлове е възможно да бъдат идентифицирани, както е описано по-долу. Чрез използване на техниката на идентификация на източника на повикване, описана в предишните части, или чрез директни инструкции от дублиращата програма може да се осигури повикването на отворен файл да бъде извършено от дублиращото искане и ако е така, таблицата на отворените файлове може временно да бъде обработена, за да осигури команда за достъп до файла даже ако това нормално е било забранено. Щом веднъж е бил осигурен достъп, но преди прекъсващият оператор да се върне, таблицата е възстановена в предишното ѝ състояние такава, че на последващите повиквания от други ползватели нормално ще бъде отказан достъп. Заявките за четене от дублиращото искане тогава ще бъдат удовлетворени, даже и оригиналното искане да предполага то да има изключително право за използване на файла.

В MS.DOS таблицата на отворените файлове е известна като System File Table (SFT). SFT е таблица, която съдържа едно въвеждане за всеки отворен файл. Всяко въвеждане съдържа модул за достъп, удовлетворен за всеки достъп до файла. Допълнителните инструкции, които са необходими за

изменение на SFT, когато дублиращото приложение изисква достъп до отворен файл, чрез временно изменение на подходящо SFT въвеждане да се създава впечатление, че правата на достъп, които нормално биха предпазвали от достъп чрез дублираща програма, вече няма да бъдат удовлетворени. Тогава достъп ще бъде разрешаван и операторът на файла ще бъде разпределен. Преди връщане към повикващия, правата на достъп в SFT въвеждането са върнати в тяхното първоначално състояние.

SFT е описана подробно в справочника "Undocumented DOS", който беше споменат по-горе, в глава 8, страници от 465 до 469 и от 490 до 494. Специално в случая на DOS операционната система стъпките, изисквани от INT 21h прекъсващ оператор за гарантиран достъп до дублиращото приложение, са показани на фигура 5.

Съгласно фигура 5 до получаване на команда за отворен файл от вида на INT 21h повикване в стъпка 70 се извършва определяне на източника на запитване, използвайки техниките, описани в предишните раздели. Ако е така, в стъпка 72 се извършва определяне на това, дали файлът, до който е поискан достъп, е вече отворен. Ако отговорът на който и да е от предходните въпроси е НЕ, тогава програмата се връща към нормалната програма на оператор INT 21h, стъпка 74. Ако отговорът и на двата въпроса е ДА, тогава специалният код, описан на фигура 5, е изпълнен. Това се извършва, както следва.

Всяко въвеждане в SFT съдържа режим на достъп, удовлетворен за заявления процес (Access Mode Attribute). В стъпка 76 провежданият или оригиналният Access Mode Attribute на определено въвеждане в SFT е запазен. В стъпка 78 Access Mode Attribute след това е променен на Deny None състояние, така че SFT ефективно е приведено в неоперативно състояние, като при това тя не може да спре множествения достъп до отворения файл. Сега нормалният INT 21h прекъсващ оператор може да бъде извикан в стъпка 80, за да отвори файла за дублиращата програма по нормалния начин. Когато дублиращата програма е свършила с файла, оригиналът, преди запомнения Access Mode Attribute е поставен в SFT (стъпка 82).

Накрая изпълнението е върнато към запитващата програма, използвайки IRET инструкцията, която чете стека (стъпка 84).

Стъпките, дадени в приложението, предполагат, че това усъвършенстване, осигуряващо гарантиран достъп до файла, е било използвано.

Прецизният начин, по който SFT е променен, не е важен, и различни методи могат да бъдат използвани. Предпочитаният метод манипулира полетата за едновременно използване "shere-prev-sft" и "share-net-machine" (виж Undocumented DOS, страница 478). Чрез манипулиране на тези полета, например чрез изменение на стойността "deny read" до "deny none", може да се предизвика достъпът да бъде удовлетворен за дублиращата програма. Алтернативно частите на Share Utility, която е част на MS.DOS, може да бъде променена така, че да бъде удовлетворен достъпът (виж op.cit., страници 490 до 494).

Подобен метод би се използвал в среди на други операционни системи, например, Novell NetWare.

С метода съгласно фигури 3 и 4 се осигурява установяване на "добро" състояние на файла. Когато файлът е отворен за дублиране, той не може да се променя, докато дублирането се извършва. Това все пак не установява, че този файл е в "добро" състояние в момента на стартиране на дублирането. Например, друго искане би могло да запише транзакция, която включва серия от записващи операции, някои от които са били изпълнени, а някои не са били изпълнени. За дублиращото искане е възможно да се получи копие на файл, съдържащ една частична транзакция, защото методът на дублиране оперира с отворени файлове.

Изобретателят е преценил, че "добро състояние" на дублиращото копие на файла може да бъде осигурено чрез закъсняване на заявката на отворения файл от дублиращо искане до момента, в който може да бъде определено, че една транзакция не продължава. Това може лесно да се постигне, тъй като повечето мрежови операционни системи са, поради самата си същност, многос-

тъпкови, т.е. те са проектирани да извършват повече от един процес в едно и също време и възможността да се забави един процес, докато завърши друг, е едно присъщо свойство на тези операционни системи. Обикновено няма значение дали дублиращата операция е поддържана за кратко време, тъй като ползвателите обикновено не са зависими от нейното приключване, особено ако файловете могат да бъдат дублирани, докато са отворени. Закъснението на заявката до отворен файл кореспондира на закъснението на старт сигнала, който слага началото на изпълнение на програмите, описани с помощта на фигури 3 и 4.

Вторият начин е приложим, когато не са налични интегрални средства за поддържане на целостта на транзакцията. Тогава може да се използва един алтернативен начин за осигуряване целостта на транзакцията, както следва. Типично транзакциите възникват в относително кратки периоди от време, с дълговременни паузи между транзакциите. Обикновено една приложна програма провежда всички операции, необходими за единична транзакция в бърза последователност, тъй като в противен случай има опасност, поради отпадане на захранването или друго смущение, да се предизвика данните върху диска да съдържат една частична транзакция, която, когато системата е рестартирана, да бъде показана като разрушен файл. Ясно е, че колкото по-кратък е периодът, за който транзакцията се записва върху диска, толкова по-малко вероятно е по време на транзакцията да се случи един страничен фатален проблем (отпадане на захранване). Чрез измерване на периода от време между успешните операции запис е възможно да се направи оценяване на стойността на транзакционалното състояние на файла. Например, ако не се извършат записи в, да кажем, по-малко от една минута, има висока степен на вероятност едно искане да е записало частично една транзакция. Чрез емпирично регулиране на периода на неактивност спрямо характеристиките на заявките може да бъде осигурен относително висок индикатор на достоверност на целостта на транзакция.

Описани са различни усъвършенствания на начините, илюстрирани на фигури 3 и 4, а именно, различни техники за иденти-

фициране на източника на заявка, техники за осигуряване на гарантиран достъп до файла, техники за осигуряване на това файлът да бъде дублиран в "добро" състояние преди започване на дублирането. Тези различни усъвършенствания се използват за предпочитане в комбинация. Те могат да бъдат използвани селективно и независимо, като не е необходимо всички те да се осъществяват едновременно в едно и също искане.

Изобретението е описано първо в условията на DOS операционна система, по-специално MS.DOS. Изобретението не се основава на определена архитектура компютри, за да може да се осъществи. То не е ограничено до DOS операционните системи и би могло да се осъществи в Novell NetWare, OS/2 или UNIX, както и на компютри Макинтош, които използват компютри от сериите 6800 и PowerPC.

Приложение

Избор на код на дублиращата програма:

- код на програма преди повикване
6F00:0009 INT 21h Повикване INT 21h
към отворен файл

6F00:000B ? Прекъсване връщане тук
Кода на програмата след повикване
Допълнителен INT 21h оператор съгласно изобретението:

- Предпазва регистрите от локални изменения

- Провереното повикване е повикване, което желаем да извърши прекъсване, отваряне, четене, запис и т.н. Ако не - скок към стария INT 21h оператор.

Проверка на стека за адрес на повикващия:

- rop dx Сегментът на повиквания е в dx регистъра

- rop cx Преместването на повикване е в cx регистъра

- push cx Поставяне на преместването обратно в стека

push dx Поставяне на сегмента обратно-стекът сега е възстановен

- DX:CX сега показва обратния адрес на повикващия, в този пример - 6F00:000B

- Установяване идентичността на повикващата програма чрез кода на изпитващата програма преди или след INT 21h повикване за известна сигнатура.

- скок Ако повикващият не е дублираща програма-скок към друг код

- скок 2 Ако файлът не е вече в употреба-скок към друг код.

Таблицата за обработка да направи преглед дали е разрешен достъп.

- pushf Въвеждане на флагове, за да симулира "iret"

- оператор Извикване оригиналния

10 INT 21h оператор, за да отвори файла.

Извършване на повикването и връщане тук.

Връщане на таблицата в оригиналното ѝ състояние.

15 Възстановяване на регистрите от локалните променливи.

iret Връщане към дублиращата програма.

Патентни претенции

1. Метод за достъп в компютърна система, базираща се на микропроцесор и съдържаща памет за инструкции, централно управляващо устройство и запамяващо средство, при което методът съдържа следните операции:

осъществява се дублиращ процес, извършван от споменатата компютърна система, която генерира старт сигнал и стоп сигнал и най-малко една дублираща инструкция за четене на налични данни от споменатото запамяващо средство; като се извеждат (четат) първа група инструкции, запаметени в паметта, към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запамяващото средство, а дублиращият процес се осъществява с файлове, които са отворени за достъп от инструкциите; при което се поддържа записът на данните, записани в запамяващото средство, и се свързват съответстващи части на първа запамяваща област на запамяващото средство и втора запамяваща област на същото запамяващо средство; като след получаване на старт сигнала и получаване на инструкция за четене се изпраща запитване до запис за вида на операцията за четене - за извършване на дублираща или нормална операция за четене, след което се четат данни от една от споменатите първа и втора запамяващи области към този за-

пис и същевременно се изпълнява и другата от споменатите дублираща и нормална операция за четене, четейки данни от първата запамеляваща област, характеризира се с това, че се извършва и операция за проверка на областта от паметта, от която произлиза инструкцията за четене и се идентифицира програмата, която е инициирала споменатата операция за четене.

2. Метод за достъп в компютърна система съгласно претенция 1, базираща се на микропроцесор и работеща в компютърна мрежа, характеризира се с това, че се присвояват (записват) към споменатия дублиращ процес идентификационни данни за ползвател, процес и/или задача и се проверява за съответствие на инструкцията за четене към дублиращата операция за четене и/или нормална операция за четене, отговаряща на заявената от ползвателя и/или по програмен път (по мрежата) с идентификационните данни за ползвател, процес и/или задача.

3. Метод съгласно претенция 2, характеризира се с това, че операцията за проверка на областта на паметта се осъществява чрез запитване на стека и се идентифицират местата със запаменени данни в стека и местоположението им.

4. Метод съгласно претенция 3, характеризира се с това, че операцията за проверка на областта на паметта представлява сравняване на кода на споменатите местоположения с кода, предвиден от дублиращата програма.

5. Метод съгласно претенция 3, характеризира се с това, че операцията за проверка на областта от паметта представлява идентифициране на Program Segment Prefix и на Environment String, асоциирани към програмата в споменатите места, като същевременно се идентифицира името на програмата, използваща Environment String.

6. Метод за достъп в компютърна система, базираща се на микропроцесор и съдържаща памет за инструкции, централно управляващо устройство и запамеляващо средство съгласно претенция 1, при което методът съдържа следните операции:

- осъществява се дублиращ процес, извършван от компютърната система, която генерира старт сигнал и стоп сигнал и най-

малко една дублираща инструкция за четене на налични данни от запамеляващото средство, след което

- се подават инструкции от паметта към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запамеляващото средство; и

- се използва средство за отваряне на файл, който се чете чрез нормална операция за четене, и същевременно средството забранява достъпа до споменатия файл за други операции за четене, характеризира се с това, че се блокира забраняващата функция на средството за отваряне на файла и се разрешава достъп за четене до споменатия файл при едновременно осъществяване на дублираща операция за четене и нормална операция за четене.

7. Метод съгласно претенция 6, характеризира се с това, че компютърната система съхранява една таблица на отворените файлове и забраняващата функция се блокира чрез временно обратно изменение на таблицата.

8. Метод съгласно претенция 1 и 6, характеризира се с това, че генерирането на споменатия старт сигнал се забавя до момента, в който завърши която и да е последователност от операции за запис във файла, който трябва да бъде дублиран.

9. Компютърна система съгласно претенциите от 1 до 8, базираща се на микропроцесор и съдържаща памет за инструкции, централно управляващо устройство и запамеляващо средство, при което към структурата на компютърната система са предвидени и следните средства:

- средство за осъществяване на дублиращ процес, извършван от компютърната система, която генерира старт сигнал и стоп сигнал и най-малко една дублираща инструкция за четене на налични данни от запамеляващото средство:

- средство за извеждане (четене) на първа група инструкции, запаменени в паметта, към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запамеляващото средство, а дублиращият процес се осъществява с файлове, които са отворени за достъп от инструкциите;

- средство, което поддържа записа на данните, записани в запаметяващото средство, и свързва съответстващи части на първа запаметяваща област на запаметяващото средство и втора запаметяваща област на същото запаметяващо средство;

- средство, което след получаване на споменатия старт сигнал и получаване на инструкцията за четене изпраща запитване до записа за вида на операцията за четене - за извършване на дублираща или нормална операция за четене, след което чете данни от една от споменатите първа и втора запаметяваща област към този запис и същевременно изпълнява и другата от споменатите дублираща и нормална операция за четене, четейки данни от първата запаметяваща област, характеризира се с това, че има и средство, което извършва и операция за проверка на областта от паметта, от която произлиза инструкцията за четене и идентифицира програмата, която е иницирала операцията за четене.

10. Компютърна система съгласно претенция 9, базираща се на микропроцесор и работеща в компютърна мрежа, характеризира се с това, че има средство за присвояване (записване) към споменатия дублиращ процес на идентификационни данни за ползвател, процес и/или задача и за проверка за съответствие на споменатата инструкция за четене към дублиращата операция за четене и/или нормална операция за четене, отговаряща на заявената от ползвателя и/или по програмен път (по мрежата) със споменатите идентификационни данни за ползвател, процес и/или задача.

11. Компютърна система съгласно претенция 10, характеризира се с това, че в структурата ѝ е предвидено средство за осъществяване на операцията за проверка на областта на паметта чрез запитване на стека и идентифициране на местата със запаметени данни в стека и местоположението им.

12. Компютърна система съгласно претенция 11, характеризира се с това, че в структурата ѝ е предвидено средство за осъществяване на операцията за проверка на областта на паметта, което сравнява кода на споменатите местоположения с кода, предвиден от дублиращата програма.

13. Компютърна система съгласно претенция 11, характеризира се с това, че в структурата ѝ е предвидено средство за осъществяване на споменатата операция за проверка на областта от паметта, което идентифицира Program Segment Prefix и Environment String, асоциирани към програмата в споменатите места, като същевременно идентифицира името на програмата, използваща Environment String.

14. Компютърна система, базираща се на микропроцесор и съдържаща памет за инструкции, централно управляващо устройство и запаметяващо средство съгласно претенция 9, която съдържа следните средства:

- средство за осъществяване на дублиращ процес, извършван от компютърната система, която генерира старт сигнал и стоп сигнал и най-малко една дублираща инструкция за четене на налични данни от запаметяващото средство;

- средство за подаване на инструкции от паметта към централното управляващо устройство, което изпълнява най-малко една нормална операция, като чете данни от запаметяващото средство; и

- средство за отваряне на файл, който се чете чрез нормална операция за четене, и същевременно средството забранява достъпа до споменатия файл за други операции за четене, характеризира се с това, че има и

- средство, което блокира забраняващата функция на средството за отваряне на файл и разрешава достъп за четене до споменатия файл при едновременно осъществяване на дублираща операция за четене и нормална операция за четене.

15. Компютърна система съгласно претенция 14, характеризира се с това, че в нея е предвидено средство за съхраняване на таблица на отворените файлове, което блокира забраняващата функция чрез временно обратно изменение на таблицата.

16. Компютърна система съгласно претенция 9 и 14, характеризира се с това, че средството за генериране на споменатия старт сигнал има структура, която забавя генерирането до момента, в който завърши която и да е последователност от операции за запис във файла, който трябва да бъде дублиран.

Приложение: 5 фигури

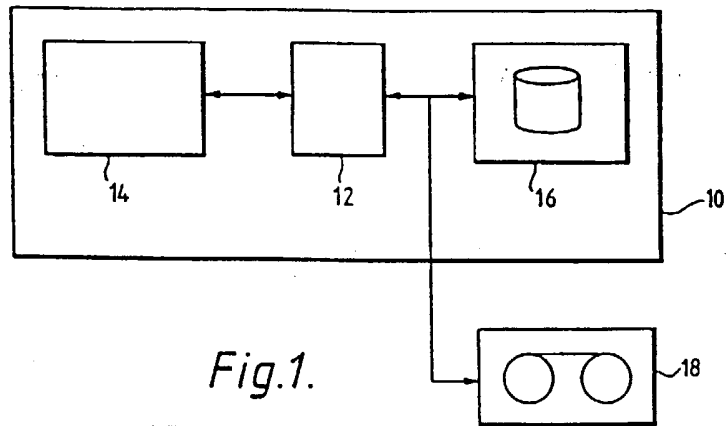


Fig.1.

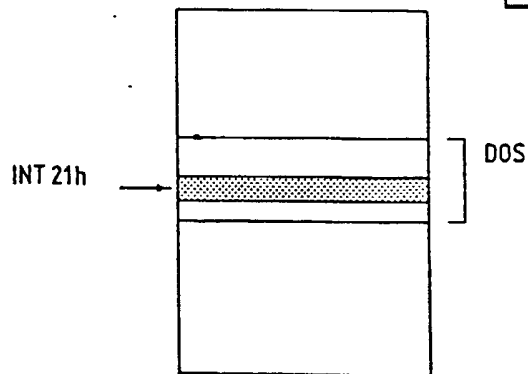


Fig.2A.

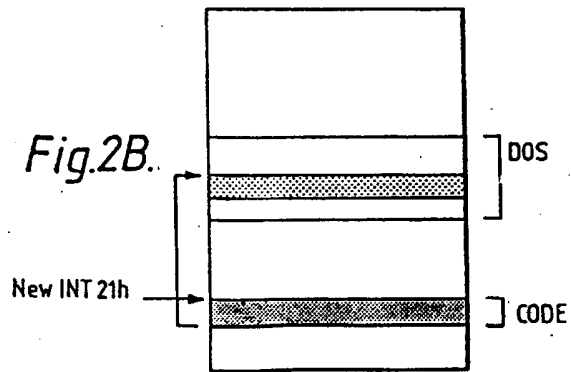


Fig.2B.

Издание на Патентното ведомство на Република България
1113 София, бул. "Д-р Г. М. Димитров" 52-Б

Експерт: К.Лазарова

Редактор: В.Алтаванова

Пор. № 40927

Тираж: 40 СР

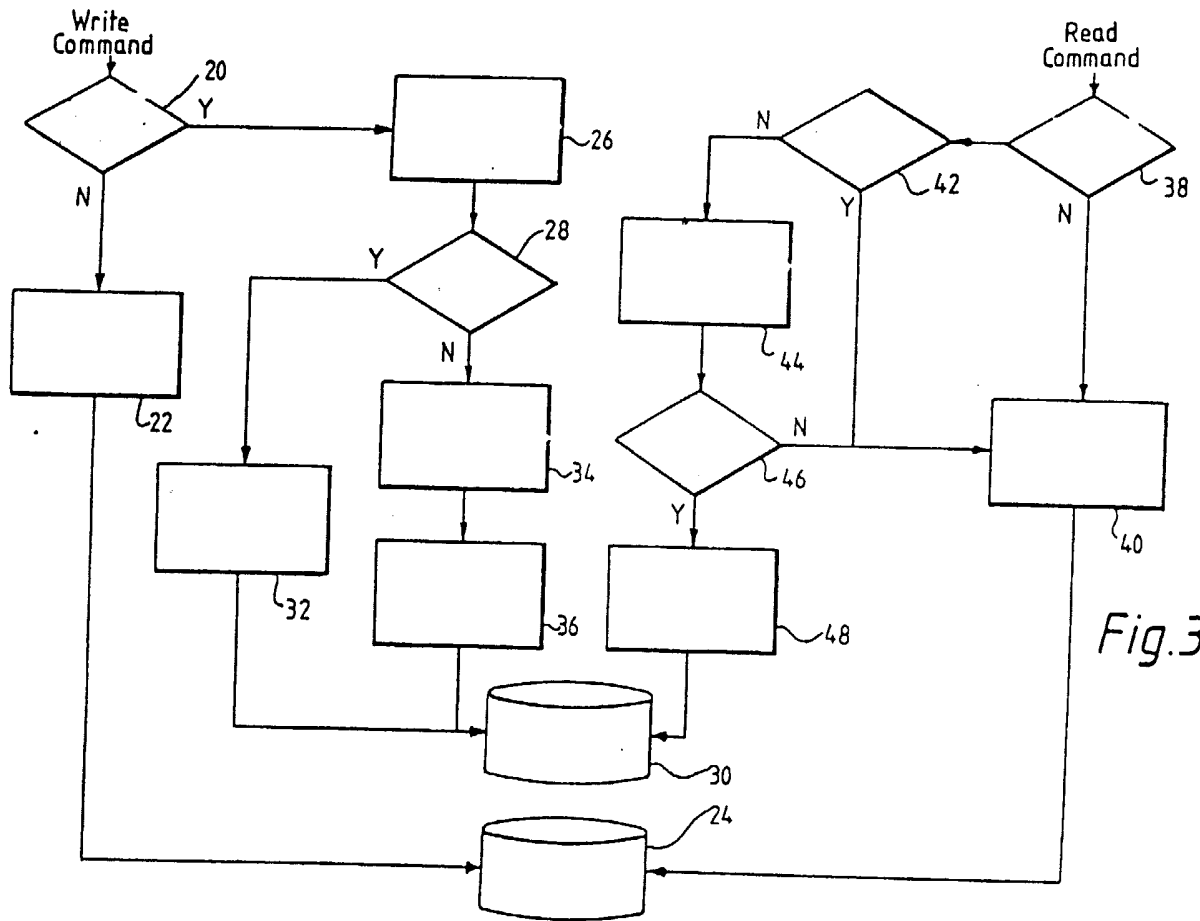


Fig. 3.

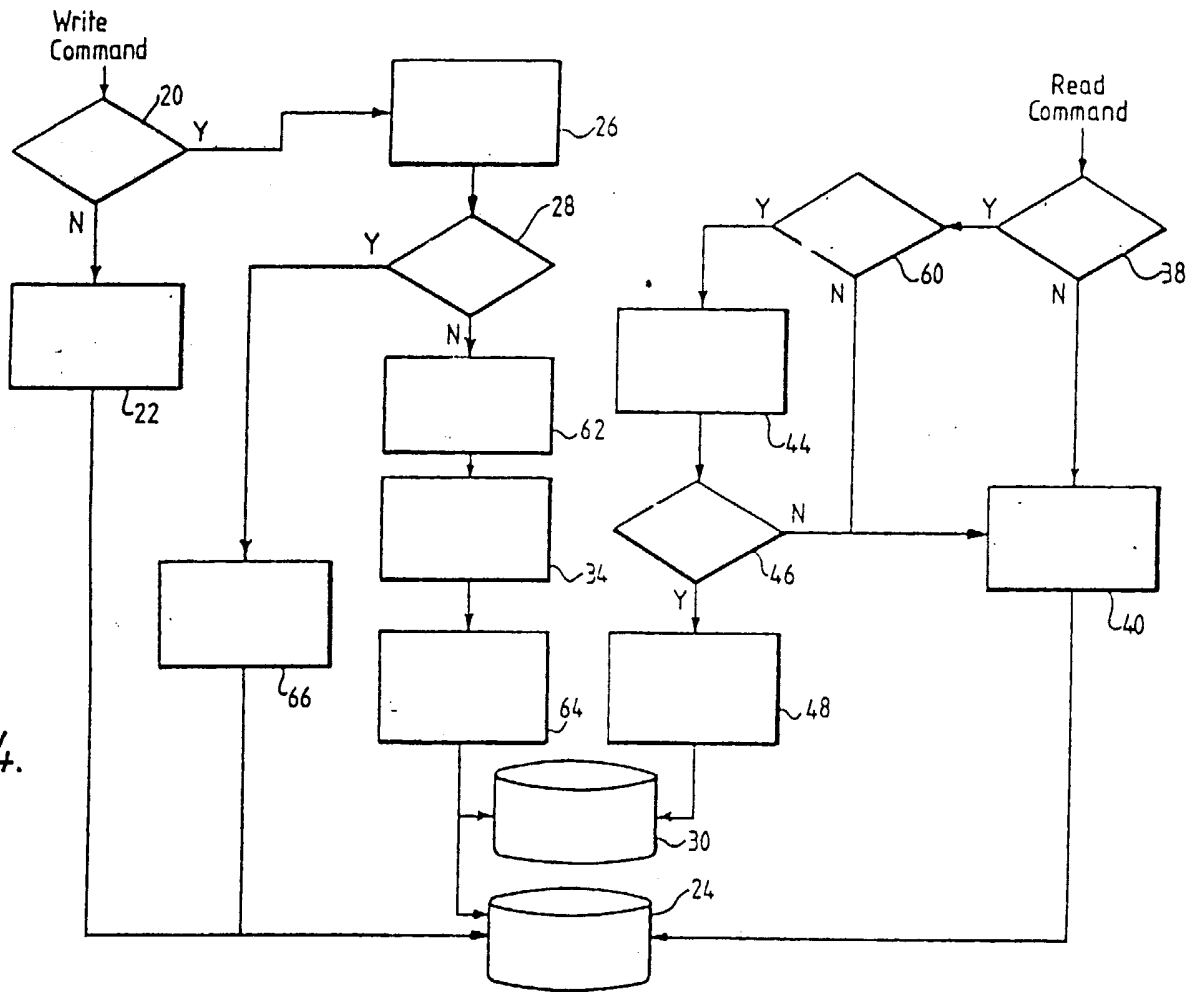


Fig. 4.

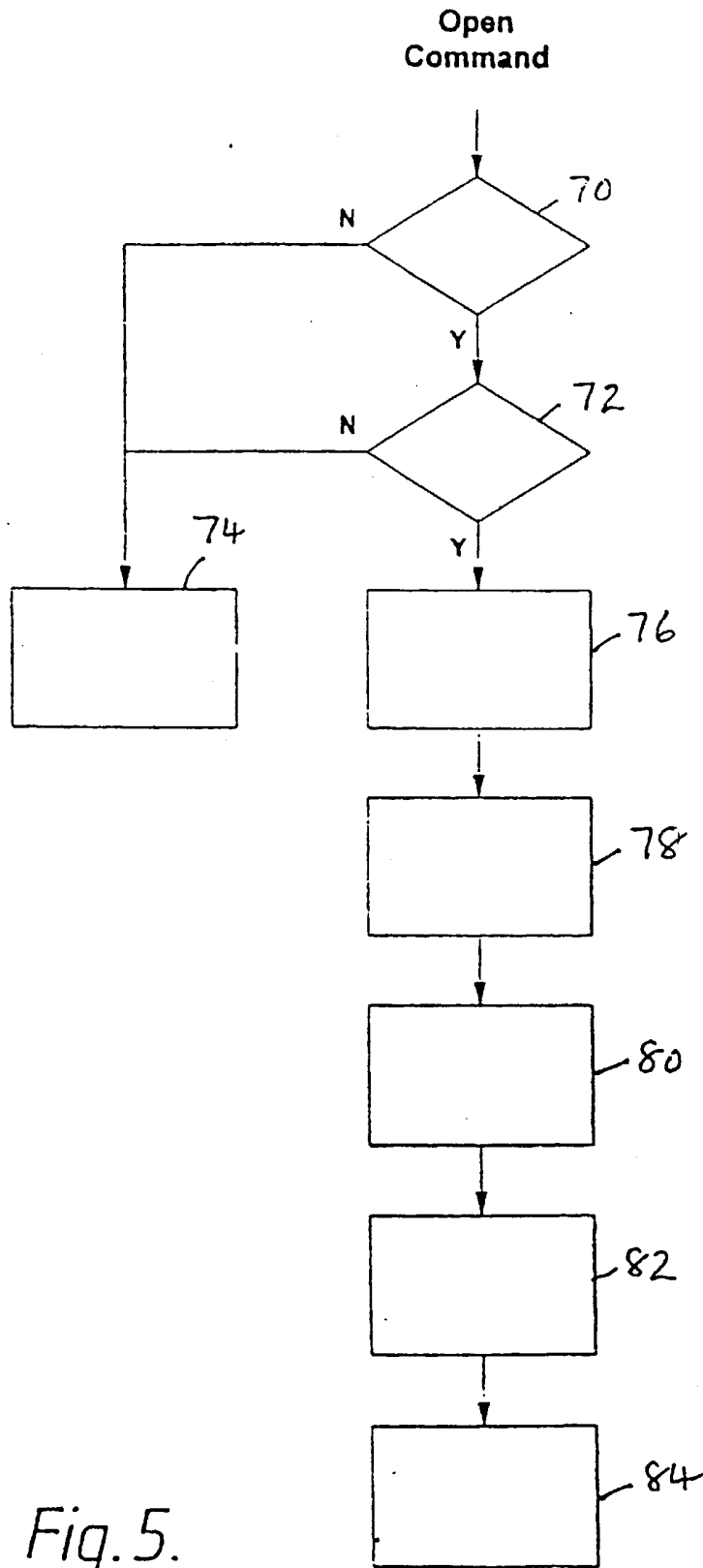


Fig. 5.