

**ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС
ТАЪЛИМ ВАЗИРЛИГИ**

**БУХОРО ОЗИҚ - ОВҚАТ ВА ЕНГИЛ САНОАТ
ТЕХНОЛОГИЯСИ ИНСТИТУТИ**

“Касбий таълим” факултети

«Информатика ва ахборот технологиялари» кафедраси

Р Е Ф Е Р А Т

Мавзу: ФАЙЛЛАР БИЛАН ИШЛАШ

Кириш.

Янги аср бўсағасида информатика фанининг гуллаб яшнаб бориши билан биргаликда тизимли дастурлаш фанини ҳам кенг ривожланиши ва амалиётда қўллаш муаммоси сезилиб қолмоқда. Чунки информатика фанининг деярли барча соҳалари тизимли дастурлашнинг йўналишлари ва усулларига таянган. Компьютер олами иккига бўлинади: аппаратли тизим қисми ва дастурий қисм. Дастурий воситаларнинг ўзи ҳам тизимли ва тизимсиз турларга бўлинади. Тизимли дастур воситалари орқали барча аппарат воситаларини бошқариш имконияти мавжуд. Айнан шунинг учун ҳам тизимли дастурлаш информатика фанининг ҳамда янги технологияларнинг ривожланишига улкан ҳисса қўшади. Йиллар ўтган сайин кўплаб янги аппарат тизимлари ихтиро қилинмоқда. Ушбу машиналар мантиқий ва амалий имкониятлари анча юқори бўлсада, аммо уларнинг асосини ташкил этувчи дастурий тизимлар ўзгартирилади.

Тизимли дастурлаш ишлаб чиқариш, табиий, меҳнат ресурслари, технологик усуллар ва илмий билимларнинг мажмуи билан ўзаро яқиндан ҳамкорликда бўлади.

Фан техника ютуқларини ишлаб чиқаришга тадбиқ этиш ишлаб чиқариш ва инфратузулмада чуқур сифатий ўзгаришларга олиб келади; ишлаб чиқаришнинг кўламлари ўсади; унинг жамланиши, ихтисослашуви ва кооперациялашуви кучаяди; нисбатан юқори навли маҳсулотларини катта кўламларда ишлаб чиқариш учун ускуна ва иншоотларига сармоя киритиш ҳажми ўсади. Бундай шароитларда ресурслар сарфлаш йўналишларини тўғри танлаш керак, чунки бу соҳага йўл қўйилган хатоларни кўп йиллар тузатиш мумкин.

Демак, ишлаб чиқаришни ривожлантириш, замонавий ахборот технологияларни ишлаб чиқиш учун янгидан янги дастурий воситалар ишлаб чиқиш ҳам ҳозирги давр талаби десак ҳеч қандай хато бўлмайди.

ФАЙЛЛАР БИЛАН ИШЛАШ.

Операцион система сифатида DOS асосий вазифаси файллаар ва дректориялар тўплами сифатида дискларни очишни ташкил этишдир. DOS файл системасининг фақат бир турини - FAT ва 7.0 весиядан бошлаб унинг файллаар узин номлари билан VEAT модификациясини қўллайди. MS DOS 1.0 да тавсия этилган файллаар билан ишлаш функциялар бошланғич тўплами жуда ноқулай эди: ҳар бир очилган файл FCB 37- байтли структураси билан таърифланиб, унинг адреси ҳамма файл операцияларда талаб этилган, маълумотлар узатиш эса DTA маълумотлар структураси орқали узатилгач амалга оширилган (маълумотлар узатиш соҳаси). MS DOS 2.0 да FAT такомиллаштирилиши билан бирга UNIX- ўхшаш тфайллаар билан ишлаш функциялар тўплами пайдо бўлиб, у файлни таърифлаш учун фақат бир 16- байтли сонни, файл ёки ускуна идентификаторни қўллаган. Файллаар билан ишлаш қолган ҳамма функциялар фақат шу сонни қуллайди. Биринчи 5 идентификаторлар система томонидан шундай инициализацияланади:

- 0: STDIN - киритишнинг стандарт қурилмаси (одатда клавиатура).
- 1: STDOUT - Чиқаришнинг стандарт қурилмаси (одатда экран).
- 2: STDERR - Хатолар тўғрисида маълумот чиқариш қурилмаси (доимий экран).
- 3: AUX - Изчил порт (одатда COM1).
- 4: PRN - Паралел порт (одатда LDT1);

Шунинг учун файллаарнинг ўқиши / ёзиши функцияларини (шунингдек буферларни дискка ташлаш) қурилмаларига ҳам йуллаш мумкин.

ФАЙЛЛАРНИ ЯРАТИШ ВА ОЧИШ.

DOS 3Ch функция: файлни яратиш.

Кириш : AH=3Ch

CX= файл атрибути.

Бит 7: Novell Netware да турли жараёнларга файлларни очиш мумкин .

Бит 6: кулланмайди.

Бит 5: архив бити (1, файл сақланмаган бўлса).

Бит 4: деректория (3Ch функция учун 0 бўлими керак).

Бит 3: том белгиси (3Ch функция томонидан инкор этилади).

Бит 2: система файли.

Бит 1: яширин файл.

Бит 0: фақат ўқиш учун файл.

DS:DX=ASCIZ адреси – файл тўлиқ номи бўлган сатр (ASCIZ ASCII- ноль билан тугайдиган белгилар сатри).

Чиқиш : CF=0 ва AX= хато бўлмаган булса, файл идентификатори.

CF=1 ва AX=03h, йўл топилмаган бўлса.

CF=1 ва AX=04h, жуда кўп очиқ файллар бўлса.

CF=1 ва AX=05h, очиш ман этилган бўлса.

Файл мавжуд булса, 3Ch функция унга ноль узинлик бериб барибир уни очади. Бу рўй бермаслиги учун 5Bh функциядан фойдаланиш керак. DOS 3Dh функция: мавжуд файлни очиш.

Кириш : AH=3Dh

AL= очиш тартиби.

Бит 0: ўқиш учун очиш.

Бит 1: ёзиш учун очиш.

Бит 2-3: резервланган (0).

Битлар 6-4: бошқа жараёнлар учун очиш тартиби.

000: мослик тартиби (мослик тартибида бу файлни бошқа жараёнлар ҳам очиши керак).

001: ҳамма операциялар ман этилган.

010: ёзиш ман этилган.

011: ўқиш ман этилган.

100: ман этилганлар йўқ.

Бит 7: туғилувчи жараёнлар томонидан файл мерос қилиб олинмайди.

DS:DX - ASCIIZ адреси- файлнинг тўлиқ номи сатрларнинг адреси.

CL- файллар атрибутлари маскаси.

Чиқиш: CF=0 ва AX=хато рўй берган бўлса, файл индикатори.

CF=1 ва AX=хатолик коди (02h- файл топилмади, 03h- йўл топилмади, 04h- жуда кўп очик файллар, 05h- очиш ман этилади, 0Ch- очишнинг нотўғри тартиби).

Функция DOS 5Bh: янги файлларни яратиш ва очиш.

Кириш: Ah=5Bh.

CX= файл атрибути

DS:DX=ASCIIZ файл тўлиқ исми бўлган адреси.

Чиқиш: CF=0 ва AX=хатолик рўй бермаган бўлса мослик тартибида ўқиш / ёзиш учун очик файл индикатори.

CF=1 ва AX= хатолик коди (03h – йўл топилмаган, 04h –очик файллар жуда кўп, 05h – очиш ман этилган, 05h файл мавжуд).

Функция DOS 5Ah: фактинча файлни яратиш ва очиш.

Кириш: Ah=5Ah

CX= файл атрибути.

DS:DX= ASCIIZ адреси – белги ва охирида ўн уч нолли битлар билан тугайдиган йўлли сатр адреси.

Чиқиш: CF=0 ва AX= хатолик рўй бермаган бўлса мослик тартибида ўқиш / ёзиш учун очик файл идентификатори. (DS:DX адрес бўйича сатрга файл номи ёзилади).

CF=1 ва AX= хатолик коди (03h- йўл топилмади, 4h- жуда кўп очик файл, 05h- очиш ман этилган).

5Ah функция ноёб исмли файлни яратади, аслида бу вақтинчалик эмас. Бундай файлни махсус олиб ташлаш керак, шунинг учун унинг номи DS:DX сатрига ёзилади. Ҳамма ҳолларда файлнинг тўлиқ номи бўлган сатр турдаги кўринишга эга, бунда диск ёки йул туширилган булса, уларнинг жорий киймати қўлланади. Файллар узин номлари билан ишлаш учун DOS 7.0 ва каттасида DOS 7.1h функция каби чақирилувчи қўшимча функциялар қўлланади.

Функция LFN 6Ch: узин номли файлни яратиш ёки очиш.

Кириш: AX=716Ch.

BX=WINDOWS 95 очиш тартиби.

Битлар 2-0: очиш.

000 = фақат ўқиш учун.

001 = фақат ёзиш учун.

010 = ўқиш ва ёзиш учун .

100 = фақат ўқиш учун , файлга охириги мурожаат

вақтинўзгартирмаслик.

Битлар 6-4: бошқа жараёнлар учун очиш /3Dh функцияни қараш.

Бит 7: туғдирилувчи жараёнлар томонидан файл мерос қилиб олинмайди.

Бит 8: маълумотлар буферланмайди.

Бит 9: файл системасининг архивлаштирилиши қўлланса файлни архивлантирмаслик.

Бит 10: файл қисқа ном билан охирида ёзиш учун D1 да сонни қўллаш.

Бит 13: қалтис хатоларда 24h тузилишини чақирмаслик

Бит 14: файлга ҳар бир ёзувдан кейин дискда буферларни ташаш.

CX = файл атрибути

DX= ҳаракат

Бит 0: файлни очиш (файл мавжуд бўлса файл хато)

Бит 1: файлни алмаштириши (файл мавжуд бўлмаса хато)

Бит 4: файлни яратиш (файл мавжуд бўлса хато)

DS: SI=ASCIZ адреси файл номи бўлган сатр

DI = файл номи қисқа варианты охирида ёзиладиган сон

Чиқиш : CF=0

AX= файл идентификатори

CX=1 файл очик бўлса

CX=2 файл яратилган бўлса

CX=3 файл алмаштирилган бўлса

CF=1 хатолик рўй берган бўлса

AX= хатолик коди (функция қўланмаса, 7100H)

Файлларни очиш функцияси “ жуда кўп очик файллар” хатоликни қайтарса (AX=4), 67H функция ёрдамида йўл қўйилувчи идентификаторлар сонини ошириш лозим.

Функция DOS 67H:файл идентификаторлари максимал сонини ўзгартириш.

Кириш: AH=67H

BX= идентификаторларларнинг янги максимал сони (20 – 65 535).

Чиқиш: CF=0, хатолик рўй бермаган бўлса

CF=1 ва AX= хатолик рўй берган бўлса, хатолик коди (масалан: 04H берилган сон очик файллар сонидан кам бўлса, агар DOS идентификаторлар янги жадвали учун етмаса)

Эсда тутиш кераки, ҳамма ички жараёнлар фақат биричи 20та идентификаторларни меърос қилиб олади ва уларга кўпроқ талаб этилса, 67H функцияни узлари чақириши керак.

ЎҚИШ ВА ФАЙЛГА ЁЗИШ.

Функция DOS 3Fh файлдан ёки қурилмадан ўқиш.

Кириш: AH=3Fh

BX= идентификатори

CX= баётлар сони.

DS:DX= маълумотлар қабул қилиш учун буфер адреси.

Кириш: CF=0 ва AX= хатолик бўлмаган бўлса, қўйилган баётлар сони.

CF=1 ва AX=05H, очиш ман этилган бўлса, 06H, нотўғри идентификатор бўлса.

Файлдан ўқишда AX да ҳақиқий ўқилган байтлар сони CX га буюрилган сондан кам бўлса, файлохирига эришилган. Ҳарбир бир кейинги ўқиши ва ёзиши операцияси файл бошидан эмас, олдинги ўқиш / ёзиш операцияси тўхтаган байтдан бошланади. Файлнинг ихтиёрий жойи ўқилиши (ёки ёзилиши) талаб этилса, 42H функция қўланади.

DOS 42H функция: ўқиш, ёзиш кўрсаткичини бошқа жойга кўчириш.

Кириш: AH=42H

BX= идентификатор.

CX: DX= кўрсаткични кўчириш керак бўлган масофага (белги билан).

AL= силжиш.

0= файл бошидан.

1= жорий позициядан.

2= файл охиридан.

Чиқиш: CF= 0 ва CX: DX= хатолик рўй бермаган бўлса, кўрсаткичининг янги қиймати (байтларда файл бошидан).

CF1= ва AX= 06H, нотўғри идентификатор бўлса кўрсаткични файлнинг реал чегараларидан ташқари ўрнатиши мумкин: манфий сонга, ундан кейинги ўқиш, ёзиш операцияси хатоликни чақиради; файл

узинлигидан каттароқ муносабат сонга, ундан навбатдаги ёзиш операцияси файл ўлчамини катталаштиради. Бу функция купинча файл узинлигини аниқлаш учун ҳам қулланади- уни $CX=0$, $DX=0$, $AL=2$ чақириш кифоя ва $CX:DX$ да файл узинлиги байтларда қайтарилади.

DOS 40h функция. Файлга ёки қурилмага ёзиш.

Кириш: $AH=40h$

BX = идинтификатор.

CX = байтлар сони.

$DS:DX$ = маълумотлар буфери адреси.

Чиқиш: $CF=0$ ва AX = хатолик руй бермаган булса, ёзилган байтлар сони .

$CF=1$ ва $AX=0.5h$ очиш ман этилган булса; $0.6h$, нотўғри идинтификатор бўлса.

Файлга ёзишда $CX=0$ кўрсатилса, у кўрсаткичи жорий киймати бўйича кесилади. Аслида DOS буферга ёзув юз беради, ундан маълумотлар ёпиш ёки улар микдори диск сектори ўлчамидан ошганда дискка ташланади.

Буфферни дархол тозалаш учун 68H функцияни қўллаш мумкин.

DOS 68H функция. DOS файл буфферни дискка ташлаш.

Кириш: $AH=68H$

BX = идинфикатор.

Чиқиш: $CF=0$ операция бажарилган бўлса.

$CF=1$ хатолик рўй берган бўлса (AX =хоталик коди).

Дастурларнинг қалтис жойлари учун самаралироқ ODH функцияни қўллаш афзалроқ.

DOS ODH функция ҳамма файл буферларини дискка ташлаш.

Кириш: $AH=ODH$

Чиқиш: ҳеч қандай.

Файлни ёпиш ва олиб ташлаш.

DOS 3EH функция. Файлни ёпиш.

Кириш: AH=3EH

BX= идентификатор

Чиқиш: CF=0, хатолик рўй бермаган бўлса.

CF=1 ва AX=6, нотўғри идентификатор бўлса.

Файл ёзиш учун очилган бўлса, ҳамма файл буферлари дискга ташланади, файл модификацияси вақти ўрнатилади ва унинг янги узунлиги ёзилади. DOS 41H функция. Файлни олиб ташлаш.

Кириш: AH=41H

DS:DX=ASCIZ адрес-файл тўлиқ номи бўлган сатр адреси.

Чиқиш: CF=0 файл олиб ташланган бўлса.

CF=1 ва AH=02H, файл топилмаган бўлса; 03H,йўл топилган бўлса; 05H очиш ман этилган бўлса.

Файл ёпилгандан кейингина уни олиб ташлаш мумкин, акс холда DOS мавжуд бўлмаган файлга ёзишнинг бажарилишини давом этади, бу файл системасини бузилишига олиб келиши мумкин. 41H функция маскаларини бирданига бир неча файлни олиб ташлаш учун қўллашга рухсат бермайди, бунга уни хужжатлаштирилмаган 5D00H функция орқали чақирилиши билан ҳам эришиш мумкин бўлса ҳам бироқ DOS 7.0 дан бошлаб файлни олиб ташлаш расмий функцияси бирданига бир неча файллар билан ишлашга қодир.

LFN 41P функция. Узун номли файлларни олиб ташлаш.

Кириш: AX =7141H

DS:DX= файлнинг узун номи бўлган сатр адреси ASCIZ

SI=0000H: маскalar рухсат этилмаган вақт CX да атрибутлар адреси рад этилади.

SI=0001H маскalar файл номида ва атрибутлар CX да рухсат этилган.

CL= файллар эга бўлиши мумкин бўлган атрибутлар.

CH=файллар эга бўлиши керак бўлган атрибутлар.

Чиқиш: CF=0, агар файл ёки файллар олиб ташланган бўлса.

CF=1 ва AX=хатолик рўй берган бўлса, хатолик коди. 7100H код бу функция қўлланмаслигини ағлатади.

Файлларни излаш.

Дискда керакли файлни топиш уни очишдан анча мураккаброқ-бунинг учун қисқа номлар билан ишлаш учун икки функция(биринчи файлни топиш ва кейинги файлни топиш) ва узун номлар билан DOS 7.0 0 да ишлашда уч функция талаб этилади (биринчи файлни топиш, кейинги файлни топиш, излашни тўхтатиш).

DOS 4EH функция. Биринчи файлни топиш.

Кириш: AH=4Eh.

AL APPEND функциясига мурожат этишда қўлланади.

CX= файл эса булиши керак булган атрибутлар (битлар 0) фақат ўқиш учун ва 5 (архив бити) рад этилади. Агар бит 3 (том белгиси) ўрнатилган бўлса, қолган ҳамма битлар рад этилади.

DS:DX= йулни ёпиши мумкин булган файл номи бўлган сатр ва излаш учун маска адреси ASCIZ.

Чиқиш: CF=0 ва файл топилса, DTA соҳаси маълумотлар адреси билан тўлади.

CF=1 ва AX=02h файл топилмаган бўлса; 03h йул топилмаган бўлса; 12h очиш нотўғри тартиби бўлса;

Бу функцияни чақириш DTA хотира соҳасини маълумотлар билан тулдиради, у индамаганда 0080h силжишдан PSP маълумотлар блоки бошидан бошланади. Лекин уни 1Ah функция ёрдамида қайта аниқлаш мумкин.

DOS 1Ah функция DTA соҳасини ўрнатади.

Кириш: AH=1Ah

DS:DX=DTA боши адреси (128 байтли буфер).

Файллар излаш функциялари DTA ни қуйидагича тулдиради.

+ 00h: байт – 0 – битлар: ASCII диск ҳарфи коди. Бит 7: тармоқ диски.

+ 01h: 11 байт – излаш маскаси (йулсиз).

+ 0Ch: байт излаш учун атрибутлар

+ 0Dh: сўз – директорияда файл тартиби рақами.

+ 0Fh: сўз – ташқи дректория бошланиши кластер номери.

+ 11h: 4 байт – резевланган.

+ 15h: байт – топилган файл атрибути.

+ 16h: сўз - DOS форматда файл яратиш вақти.

Битлар 15-11: соат (0-23).

Битлар 10-5: дақиқа.

Битлар 4-0: 2 бўлинган сония номери (0-30).

+ 18h: сўз – DOS форматда файл яратиш санаси.

Битлар 15-9: йил, 1980 дан бошлаб.

Битлар 8-5: ой.

Битлар 4-0: кун.

+ 1Ah: 4 байт – файл ўлчами

+ 1Eh: 13 байт – ASCIIZ – кенгайтирилиши топилган файл номи.

DTA маълумотлар билан тўлдирилган, излашни давом эттириш учун хатолик қаайтарилмагунча 4Fh функцияни чақирилиши керак.

DOS 4Fh функция: кейинги файлни топиш.

Кириш: AH=4Fh

DTA – 4Eh ёки 4Fh функцияни олдинги чақирувдан маълумотларни сақлайди.

Чиқиш: CF=0 ва DTA хатолик руй бермаган бўлса, кейинги топилган файл ҳақида маълумотларни сақлайди.

CF=1 ва AX= хатолик руй берган бўлса, хатолик коди.

Файллар узин номлари билган ҳолда DOS 71h функция уч функция остидан тўплаш кулланади, уларни 1FSmrg юборилган бўлсагина куллаш мумкин. LFH 4Eh функция узўн номли биринчи файлни топиш.

Кириш: AX=714 Eh.

CL= файл эга бўлиши мумкин бўлган атрибутлар (0 ва 5 битлар рад этилади).

CH= файл эга бўлиши керак бўлган атрибутлар.

SI= WINDOWS – фармат сана / вақтни куллаш.

SI= сана / вақт DOS форматни куллаш.

DS:DX= ASCIIZ адреси – излаш учун маскали сатр.

ES:DI= файл туғрисида маълумот учун 318 байтли буфер адреси.

Чиқиш: CF=0

AX= излаш идентификатори.

CX= Unicode – байроқ.

Бит 0: қайта ўзгартирилмайдиган Unicode – белгилар ўрнида узин ном остига чизишга эга.

Бит 1: қисқа ном қайта ўзгартирилмайдиган Unicode белгилар ўрнида остига чизишга эга.

CF=1 AX= хатолик рўй берган бўлса, хатолик коди (7100H – функция кулланмайди).

Маска остга ва излаш атрибутларига туғри келувчи файл топилган бўлса, маълумотлар сатҳи ES:DI адрес бўйича қуйидагича тулдирилади:

+04H : 8 байт – файл яратиш вақти

+06H : 8 байт – файлни охирги очиш вақти

+14H : 8 байт – файлнинг охирги модификацияси вақти

+1CH : 4 байт – файл узунлиги катта қўш сўзи.

+20H : 4 байт – файл узунлиги кичик қўш сўзи

+24H : 8 байт – резервланган

+2CH : 260 байт – ASCIIZ – файл номи узун

+130H : 14 байт – ASCIIZ – файл номи қисқа

Бнда яратиш (очиш) модификациялари саналари функция чақирилишида SI кийматига мос форматларнинг иккисидан бирига ёзилади. Windows – формат 64 битли сон 1 январ 1601 йилдан 100 нано секунд интервалли.

Агар DOS – формат қулланса – катта қўш сўзи DOS – сана ёзилади, кичикка DOS – вақт ёзилади.

Функция LFN 4Fh : кейинги файлни топиш.

Кириш: AX = 714Fh

BX = излаш идентификатори (4Eh функциядан).

SI = сана / вақт формати.

ES:DI = файл тўғрисида маълумот учун буфер адреси.

CF = 0 ва CX= Unicode - байроқ, кейинги файл топилган

бўлса.

CF = 1, AX= хатолик коди, хатолик рўй берган бўлса (7100h – функция қулланилмайди).

Функция LFN A1h: файл излашни тугатиш.

Кириш: AX= 71A1h

BX= излаш идентификатори.

Чиқиш: CF= 0, операция бажарилган бўлса.

CF= 1 ва AX= хатолик коди, хатолик рўй берган бўлса (7100h – функция қулланмайди).

Файллар билан ишлаш кўп функцияларни қўлловчи мисол тариқасида Н рус ҳарфларини латинча Н билан ҳамма файлларда TXT кенгайтиришлар билан жорий дректорияда алмаштирувчи дастурлар кўриб чиқамиз.

(Бундай алмаштириш Fidonet тармоқ орқали узатилувчи ҳар бир матн учун талаб этилиб, дастурий таъминоти Н рус ҳарфини бошқарувчи беги каби қабул қилади).

Русча “Н” ларни латинча “Н” билан жорий дректорияда TXT кенгайтириш

билан ҳамма файлда алмаштилади.

```
.model    tiny
.code
org       100h          ; COM- файл.
start :
    mov    ah , 4Eh      ; бирнчи файлни излаш.
    xor     cx , cx      ; системали эмас, дириктория эласвақт.
    mov     dx,offset filespec ; DS:DX да излаш учун маска.
file_open :
    int     21h
    jc      no_more_files ; CF=1 бўлса, файллар тугаган.
    mov     ax,3d02h      ; ўқиши ва ёзиш учун файлни очиш.
    mov     dx,80h+1Eh    ; DTA ни силжитишнинг файл номини
                        ; силжитиш.
    int     21h          ; DTA бошидан.
    jc      not_open      ; файл очилмаган бўлса, кейингисига
                        ; ўтиш.
                        ; кеингиси.
    mov     bx,ax         ; BX да файл идентификатори.
    mov     cx,1          ; бир байтни ўқиш.
    mov     dx,offset duffer ; буфер бошланиши – DX да.
read_next :
    mov     ah,3Fh        ; файлни ўқиш.
    int     21h
    jc      find_next     ; хатолик бўлса – кейингисига
                        ; ўтиш.
    dec     ax            ; AX= 0 бўлса – файл тугаган.
    js      find_next     ; кейингисига ўтиш.
    cmp     byte ptr buffer,8Dh;Русча “Н” ўқилмаган бўлса.
```

```

jne      read next      ; кейинги байтни ҳисоблаш.
mov      byte ptr buffer,48h; бошқача – буферга ёзиш.
                                ; лотнча “Н” ҳарфни.
mov      ax,4201h      ; файл кўрсаткичини жорий
                                позициядан.
dec      cx            ; бир тип орқага жойлаштириш.
dec      cx            ; CX= 0FFFFh
mov      dx,cx         ; DX= 0FFFFh
int      21h
mov      ah,40h        ; файлга ёзиш.
inc      cx
inc      cx            ; бир байт (CX=1)
mov      dx,cx         ; буфердан DS:DX га.
int      21h
jmp      short read_next ; кейинги байтни ҳисоблаш.
find_next :
    mov      ah,3Eh      ; олдинги файлни ёпиш.
    int      21h
not_open :
    mov      ah,4Eh      ; кейинги файлни топиш.
    mov      dx,80h      ; DTA ни PSP бошидан силжитиш.
    jmp      short file_open
no_more_files :          ; файллар тугаган бўлса.
    ret                ; дастурдан чиқиш.
filespec    db      “*.txt”,0 ; излаш учун маска.
buffer      label byte ; ўқиш / ёзиш учун буфер.
end         start      ; дастур охири орқасида.

```

Файл системасини бошқариш.

MS DOS 2.0 дан бошлаб файл системаси дректориялар кўринишида ташкил этилган. Файлларни излаш жорий директория доирасидагина бажарилади, файлларни яратиш ва олиб ташлаш эса директорияларга тўғри келмайди, энг паст даражада директория бу файл бўлиб, унинг атрибутида бит 4 бирга ўрнатилган ва у киритилган файллар номи, рўйхати, улар атрибути ва физик адресларини дискда сақласа ҳам.

Функция DOS 39h: директория яратиш.

Кириш: AH=39h

DS:DX= ASCIZ адрес – охиргисидан ташқари ҳамма директориялар мавжуд бўлган йўлли сатир. DOS3.3 ва анча эрта версиялар Учун бутун сатир узунлиги 64 байтдан ошмаслиги керак.

Чиқиш: CF=0, директория яратилган бўлса.

CF=1 ва AX=3, йўл топилмаган бўлса; 5, агар очиш ман этилган бўлса.

Функция LFN 39h: Узин номли дректория яратиш.

Кириш: AX=7139h

DS:DX=ASCIZ адреси – йўлли сатр.

Чиқиш: CF=0, дректория яратилмаган бўлса.

CF=1 ва AX= хатолик коди (7100h, функция йўлланмаса).

Функция DOS 3Ah: дректорияни олиб ташлаш.

Кириш: AH=3Ah

DS:DX=ASCIZ адрес охирги директория олиб ташланган йўлли сатр. (фақат у бўш бўлса, жорий бўлмаса, SUB ST буйруқ билан банд бўлмаса).

Чиқиш: CF=0, директория олиб ташланган бўлса CF=1 ва AX=3, йўл топилмаган бўлса; 5 очиш ман этилган бўлса, 10h – олиб ташланаётган директория жорий бўлса.

Функция : LFN 3Ah : узун номли дректорияни олиб ташлаш .

Кириш : AX=713Ah

DS: DX = йўлли сатир адрес .

CF = 0 , директория олиб ташланган бўлса , акс ҳолда CF=1
ва AX= хатолик коди .

Функция DOS 47h : жорий директорияни аниқлаш .

Кириш: AH = 47h

DL = диск номери (004h- жорий , 01h – A ваҳоказо)

DS : SI = жорий йўл учун 64 – байтли буфер . (ASCIZ –
диск номисиз , биринчи ва охирги белгисиз \ сатир) .

Чиқиш : CF = 0 ва AX =01 001h , оператця бажарилган бўлса .

CF = 1 ва AX = 0Fh , мавжуд бўлмаган диск кўрсатмаган
бўлса .

Функция LFN 47h : узун номли жорие директорияни аниқлаш .

Кириш: AX = 7147h .

DL – диск номери .

DS : SI = йўл учун буфер (ASCIZ – диск номисиз , биринчи
ва охирги белгисиз \ сатир . Фақат узун номларни сақлаш шарт эмас –
жорий директория охирги алмашувида қўлланган йўл қайтарилади) .

Чиқиш: CF = 0 , директория аниқланган бўлса CF = 1 ва AX =
хатолик коди .

Функция DOS 3Bh : директория алмаштириш .

Кириш : AH : 3Bh

DS : DX – 64 – баетли ASCIZ буфернинг жорий директория
бўлувчи йўл адреси.

Чиқиш: CF = 0 , директория ўзгартирилган бўлса, акс ҳолда CF = 1
ва AX = 3 (йўл топилмаган).

Функция LFN 3B: узун номли директорияни алмаштириш.

Кириш: AX=713BH

DS:DX= йўли ASCIZ -буфер адреси .

Чиқиш: CF=0, директория ўзгартирилган бўлса, акс ҳолда CF=1 ва AX-хатолик коди.

LFN исталган функциялари билан ишлашдан олдин бир марта 0A0H функция остини чақириб, файллар номи ва йўллар учун буферлар ўлчамини аниқлаш лозим.

Функция IEN 0A0h : VFAT файл системаси бўлимим бўйича маълумот олиш.

Кириш: AX=71F0H.

DS: DX=ASCIZ бўлим номли сатр адреси.

DS: DI=файл системаси номи учун буфер адреси.

CX= буфернинг ES : DI ўлчами (одатда 32 байт)

Чиқиш: CX=0, AX=0000H ёки 0200H.

BX= файл системаси байроқлари.

Бит 0 : ёйлаш функциялари белгилар регистрини ҳисобга олади .

Бит 1: белгилар регистри директориялар номи учун сақланади.

Бит 2: Unicode белгилари қўлланади.

Бит 14: LFN функциялари қўлланади.

Бит 15: бўлим сиқилишини киритилган.(Doreblespace).

CX = файл номи максимал узунлиги (одатда 255) .

DX = йўлнинг максимал узунлиги (одатда 260) Windowis SP1 CD – ROM учун 0000h ни қайтаради .

CF = 1 ва AX – хатолик коди , хатолик рўй берган бўлса (7100h , функця қўлланмаса) .

Бундан ташқари LFN исталган функцияни чақиришда CF ни 1 га , DOS олдинги версияларига тўғри келиши учун , ўрнатиш лозим . DOS эски версиялари CF ни ўзгартирмаган , шунинг учун натижада , функция қўлланмаса , CF 1 га тенг бўлади .

ХУЛОСА

Хулоса қилиб шуни айтиш керакки, мен ўз курс ишимни тайёрлаш давримда шу нарсани англадимки, ахборотлаштириш соҳаси биргина дастурчилар учунгина эмас, балки инсоният борки ахборот уз узидан узгариб туради. Банк соҳасими, маиший соҳами, кайси соҳа бўлишидан қатий назан ахборотни автоматлаштириш бу замон талаби бўлиб қолмоқда.

Асосан бу нарсани ишлаб чиқариш фаолиятида яккол кўзга кўриниши керак. Маркетинг фаолиятида автоматлаштириш ўзининг ижобий томонларини кўрсатмоқда. Бундан олдин бир давлат иккинчиси билан савдо олди соттиси билан шуғулланса, бу ўртада неча кишилар оввора бўлган. Хозирчи бир киши утириб бемалол ўз савдосини олиб бориши мумкин.

Мен шунга амин бўлдимки автоматлаштириш ҳамма соҳа учун ҳавадек зарурдир. Мен ҳам шу ўринда анча ўзимга керакли маълумотларни олдим.

Мен ўйлайманки келажакда бу курс ишимдан ўрганиб олган билимларим менга ижобий йулда ас қотади. Тенгдошларимга ва биздан кейинги укаларимизга шуни айтмочиманки автоматлаштириш замон талабидир.

Фойдаланилган адабиётлар рўйхати.

1. Зуков С. В. Assembler для DOS, WINDOws, UNIX.

Москва 2000.

2. ден Гукин IBMPC дея Чайников . Москва 2005 .

3. [www referat.ru](http://www.referat.ru)

4. [www r0.ru](http://www.r0.ru).

5. С.Симонович, Г. Евсеев, Программирование C++. Москва

2003 год.

6. [http:// assembler. host. net. Kg/html/lecture 9. html](http://assembler.host.net.Kg/html/lecture%209.html).

МУНДАРЖА.

Кириш.....	1
Файллар билан ишлаш.....	2
Файлларни яратиш ва очиш.....	2
Ўқиш ва файлга ёзиш.....	7
Файлни ёпиш ва олиб ташлаш.....	9
Файлни излаш.....	10
Файл системасини бошқариш.....	16
Хулоса.....	19
Фойдаланган адабиётлар рўйхати.....	20