چون تعداد cluster ها کمتر ← size harddisk کمتر ← size volume کمتر

در هر partition، 2 cluster اول از تیبل داده دارد و reserved است

(بین ۴، 8، ۱۶ یا 32 و حداکثر 64 متغیر)

size volume = $2^{32} * \left(\text{sector per cluster}\right) * \left(\text{bytes per sector}\right)$

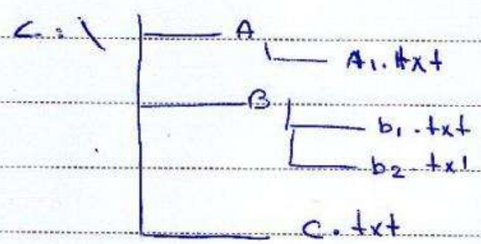
$\vdots$

2^{16} x

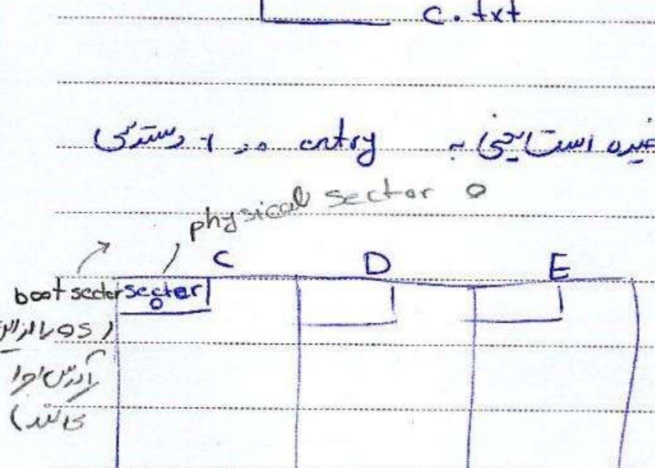
2^{12}

2^{32}

file ها و directory ها داخل volume چگونه ذخیره می‌شوند


C:\
A
A1.txt
B
b1.txt
b2.txt
c.txt

در هر volume، 2 cluster اول ذخیره است یعنی به entry ۰ و ۱ دسترسی پیدا می‌کنیم


physical sector 0
boot sector
sector 0
C
D
E
(و OS را از این آدرس اجرا می‌کند)

سنم ← OS را با چه ترتیب کند D:

Subject:
Year. Month. Date. ( )

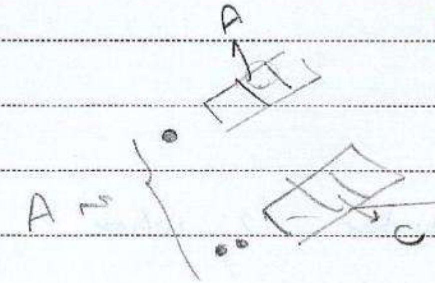
BIOS ← وقتی کامپیوتر را روشن می‌کنیم از partition ها، bootsector را می‌خواند ببیند bootable هستند یا نه اگر بود از همان جا OS را run می‌کند

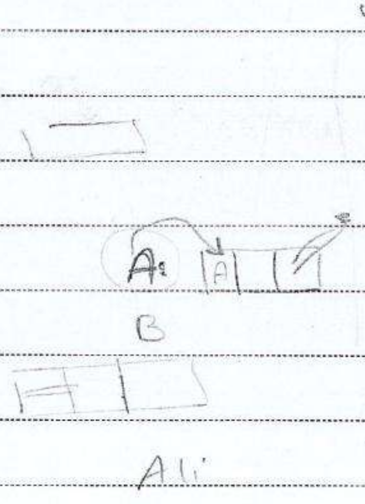
boot sector ، sice ، " ...

hard disc cluster

بعضی وقت‌ها کار با FAT خطرناک است

partition magic ← سوها
اگر وسط کار برق برود
تخلیه بعضی از ... و مشخص نیست



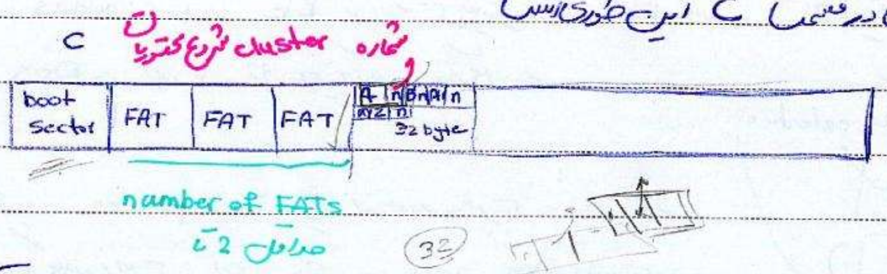


PAPCO

Subject: A A 5 (5) → 7

Year. Month. Date. ( )

هر partition بندی در قسمت C این طوری است


C
شماره cluster شروع محتویات
boot sector
FAT
FAT
FAT
A
n
B
n
A
n
xyz
n
32 byte
number of FATs
معادل 2 تا
32

از تقسیم Number of sectors in volume بر Sector per cluster نوع FAT را بدست می آوریم

اگر از 2^{12} بیشتر بود FAT16 یا FAT 32 است ...

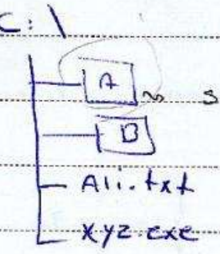
FAT 12, FAT 16 → تعداد فایل ها و شاخه های موجود در root محدود است

FAT 32 → بدون محدودیت

المان های موجود در هر جا تمام اطلاعات آن ها در data structure 32 بایتی ذخیره می شود

این subdirectory ها مثل اطلاعات آن را نگه می دارند اسم، زمان ایجاد است ...

و اولین cluster که محتویات آن را نگه می دارد


C:\
A
B
Ali.txt
xyz.exe
subdirectory / directory

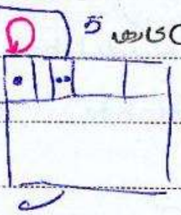
ولی دیگر sub directory ها ساختاری شبیه همین مدل directory ساختاری می شوند

داخل آن cluster به هم اطلاعات 32 بایت 32 بایتی ذخیره می شود مثلاً cluster جلوی folder A

• به خودش (مسیر جاری) که شماره اولین cluster آن cluster را در خودش نگه می دارد → به A اشاره می کند

•• به cluster پدرش ← اگر root directory باشد cluster آن صفر را نشان می دهد 5

مثلاً جلوی A، 5 است



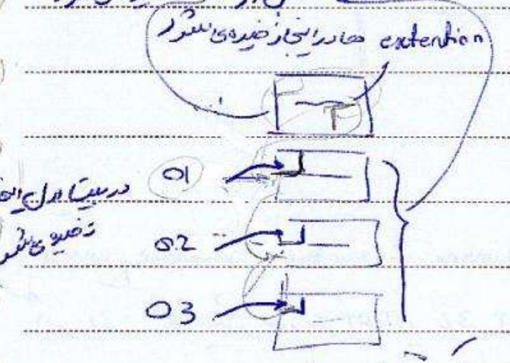
داخل folder چون file نیست محتویات وجود ندارد

بلکه cluster های که محتویات المان های داخل folder را دارند در خود نگه می دارد

به جز ساختار directory این . و .. وجود دارد و علت آن این است که می خواهیم از از

یک subdirectory به راحتی به پدرش برگردیم (در DOS)

وقتی windows از file سایزش بیشتر از 255 باشد، 32 بایت اول را می‌گیرد در همان جا مثل DOS بعد چند تا 32 بایت دیگر در نظر می‌گیرد ← قبل از آن ذخیره می‌شود


extention ها در اینجا ذخیره می‌شود
01
02
03
در بیت اول اینها ذخیره می‌شود
بلافاصله بعد از آن ذخیره می‌شود

در DOS، process، 32 بایتی از حروف بین A تا Z می‌برد process می‌کند ولی 01 و 02 ـــ برای DOS معنی ندارد و در کار DOS مشکلی پیش می‌آید

در root وقتی ما یک file را delete می‌کنیم بیت اول آن file ۷ می‌خورد یعنی "deleted شد" و به جای آن E5 قرار می‌گیرد ← طرز کار recovery ← از حالت E5 در می‌آورد ← حتماً باید cluster های آن overwrite نشده باشد. 8 بایت اول اسم را مشخص می‌کند و فقط حرف اول اسم از بین می‌رود.

اگر خود کاراکتر E5 را بخواهیم در خودش باشد 05 به جای E5 قرار می‌دهیم

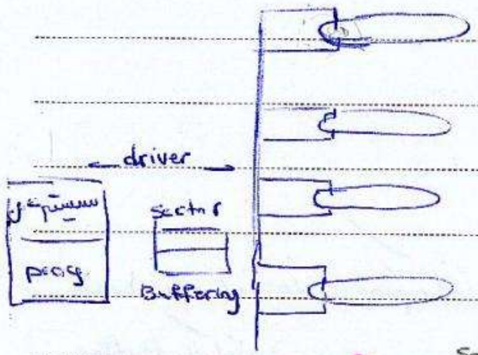
در C و ... با absread می‌توانیم اطلاعات یک cluster / sector را می‌خواند به ما نشان می‌دهد

این 32 بایتی‌ها
File
Folder
volume name ← اسم ~~File~~ partition label

در drive یک label دارد که در یک data structure، 32 بایتی ذخیره می‌کند با اسم و extention اگر این باشد در Attribute ← به عنوان volume name است

label هم در root آن ذخیره می‌شود.

Disk Access

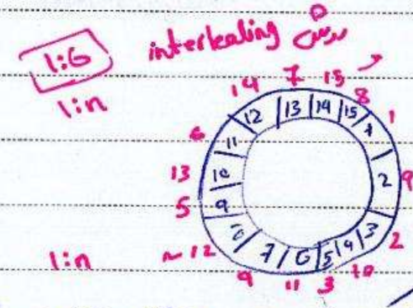


(Cylinder, Head, Sector)

(۰ , ۰ , ۶)

اولین sector هر درایو روی partition است

درخواست به این: hard disk می‌خوانیم که کدام cylinder ، sector
در track باشد



هرچه به مرکز نزدیک‌تر شویم میزانی که بچرخیم سکتور متفاوت است ولی حجم اطلاعات باید یکسان باشد در مرکز چگال‌تر است یعنی در مرکز اطلاعات بیشتری باید در واحد سطح قرار بگیرد.

در هنگام خواندن وقتی به آخر ۱ برسیم چون یک مقدار کمی زمان دارد by default یک مقداری دور جلوتر است بعد است چون مقداری data کم کردیم که تا برگردانیم عقب

② یک دور باید بچرخد تا به اول برسد

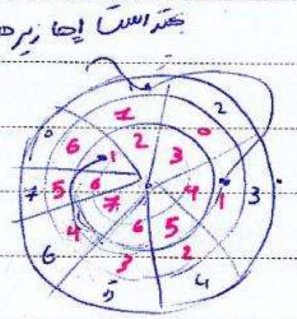
۱- روش interleaving   این روش در ۱ track است

در روش interleaving بستگی به اینکه مقدار کمی یا delay است مقدار n را مشخص می‌شود مثلاً 1:6 برای همه جواب می‌دهد. هرچه n بیشتر شود مجبور می‌شویم برای پیدا کردن track بعدی بیشتر بچرخیم اگر کمتر باشد یعنی delay کمتر است

159
22370001-5

بهتر است اینها زیر هم نباشند

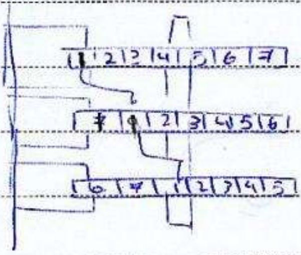
2- track skewing



اینجا هم اما چقدر فاصله بگذاریم به delay بستگی دارد.

این روش بین trackهای متوالی کارند

Cylinder skewing :3.


1 2 3 4 5 6 7
7 1 2 3 4 5 6
6 7 1 2 3 4 5

headهای توانند موازی بهم کار کنند بعضی به memory دوجهته port دارند یعنی همزمان می‌توانند
Buffer را پر کنند

B800:
40
41
42
43

برای 8

LPT1 0004:0008

]

mov AL, [

IN AL

TSR

multitasking ability

Terminate program & Stay Resident

اجرای برنامه‌های exe سپس loader در حافظه load می‌کند آنرا بعد از اجرا شدن وقتی INT 21H
function 4C00H را call می‌کنیم دیگر این حافظه بدرد نمی‌خورد.
برای مقیم شدن یک برنامه در حافظه آنرا TSR می‌نویسیم و بعد آنرا در Auto Exec.BAT قرار می‌دهیم

1- در انتهای برنامه نباید سرویس 4C00H از وقفه 21H بیست
2- باید دستوری قرار دهیم که برنامه در حافظه ماندگار شود.

INT 21H
function ← 31H

DX = اندازه‌ای حافظه‌ای که می‌خواهید در حافظه بماند >> Shift Right تا 4

```
mov   AH, 31H
mov   DX, سایز برنامه بر حسب پاراگراف  →  (DX+15) >> 4
INT   21H
```

31

می‌توانیم هی trigger کنیم تا کار را دوباره اجرا کند
که چجوری trigger کنیم ① دستی آدرس را داشته باشیم

② از طریق Interrupt ها را trigger کنیم یعنی خود INT از طریق یک device خارجی -- *
trigger می‌شود مثلا
- 08H timer
- 09H keyboard

یعنی جای این routin این کدی بنشیند که در آخر برنامه باید خود این را دهیم handle کند

جایگزینی آدرس روتین با روش ۲ | 25H / INT 21H

Fun 25 INT 21H

خواندن آدرس روتین وقفه | 35H / INT 21H

AH
AL ←
INT

Subject:
Year. Month. Date. ( )

get interrupt vector

mov AH , 35H → 　　　　　　　　OLD_ISR DW 4 DUP(?)

mov AL , 09H → interrupt number
ورودی ،،

INT 21 H → { seg = ES, offset = BX } → Interrupt routine 　　　　به routine Keyboard جایش را می‌گیریم

mov word ptr OLD_ISR , BX

mov word ptr OLD_ISR + 2 , ES

replace کنیم در حافظه {
mov AX , 25H → set interrupt vector
mov DX , offset New_ISR → offset جدید
mov AL , 09H → DS → seg جدید جای قدیم بنشیند
INT 21H
}

routine را در Interrupt‌ای که می‌خواهیم قرار بدهیم باید
مشخص کنیم که DS شروع برنامه‌ی ماست

نیازی به تعیین codeseg نداریم چون داخل همان
برنامه است

new_ISR : mov AX , Bx

new_ISR ENDP

* برنامه‌ای بنویسید با استفاده از TSR برنامه‌ای را در حافظه مقیم کند به طوریکه در هر بار اجرا شدن یک کاراکتر
(0,0) را تغییر دهد

PAPCO

Subject:
Year. Month. Date. ( )

```
Seg
main:   Jmp  INIT
        OLD_ISR DB  4DUP(?)
  New_ISR  PROC

        mov   ES,B800H
        mov   DI,01 H
        mov   AX,ES:[DI]
        NOT   AX
        mov   ES:[DI],AX
                word PTR
        Jmp   ES:OLD_ISR
  new_ISR  ENDP
```

اگر می‌خواستیم routine قبلی را هم اینجا اجرا کنیم در نقطه
کد IRET می‌گذاشتیم اما حالا نمی‌توانیم خیالمان راحت است
که حالا این به جای jmp می‌کند به INT routine در آدرس
دقتاً آخرش IRET است

```
mov  ES: WORD PTR  OLD_ISR+2
```

آدرس routin قبلی ← Jmp

نه call می‌کنیم انتقال دارد RET ببینید در جدول ما IRET داریم

یعنی Jmp ↔
RET ↔ CALL
IRET ↔ INT

INIT : ~ routine بعد از label اطلاعات آدرس

offset اینجا را از هم کم می‌کنیم

```
~~mov AH,31H~~
mov  DX,(OFFSET INIT _ OFFSET NEW_JSR)
 directive
 APP 1x,15
ENDS   SHR  DX,4
       INT  21 H
END  main
```

یک جای؟

آدرس جدید را نگه داشته می‌شود

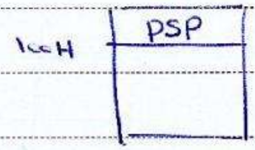
```
mov  DX,New-ISR
mov  AL,09H
mov  AH,25H
INT  21H
```

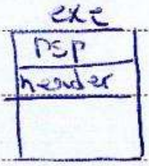
آدرس قبلی حفاظت می‌شود

```
mov  AH,35H
mov  AL,09 H
INT  21 H
mov  WORD PTR  OLD-ISR , Bx
```

متفاوت از .com می‌نویسید که header نداشته باشد ولی PSP (100byte) دارد

Subject:
Year.	Month.	Date.	( )




exe
PSP
header

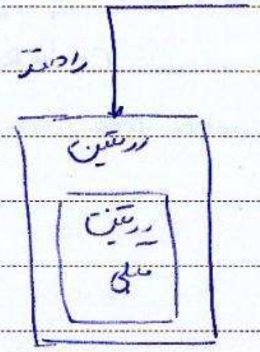
08 شماره اینتراپت تایمر

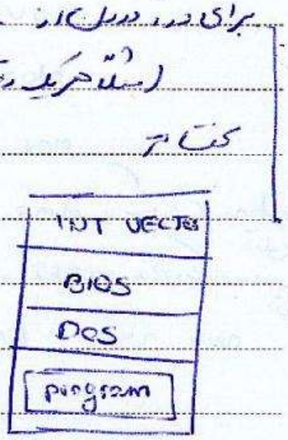
18.2 بار فراخوانی معادل 1s

برای دوباره دیدن از TSR یا دنبال کردن کار جدید به TSR

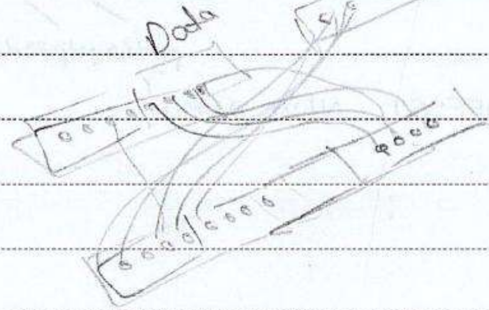
(مثلاً هر یک دقیقه رنگ را عوض کند یا 30s بوق بزند)

کنتر


را صدا
روتین
روتین
قبلی


INT VECTOR
BIOS
DOS
program

49
31
18



PAPCO

Subject:
Year. Month. Date. ( )

Disk Browser (with undelete feature)

{ Dos: Dir, cd, delete
| Windows: شبیه این نشان دهد ... , ... , Undelete

در سیستم‌های که تنظیم شدن delete کردن را replace کردن یک کار الکتریکی در حفظ به

باید cluster و Sector کار کنیم و بعد این عملیات را انجام دهیم باید جدول را برای clusterها used و unused بشه

parallel ports (پورت‌های موازی) — Direct hardware programming / DOS INT / BIOS INT

جاهایی که در windows ای ترانزیستور وصل کنیم را مجبور باشیم به assembly که بنویسیم باید این را بدانیم

connector به serial به پین بیت از تک سیم بر زمان‌های مختلف در زمان بیت serial بفرست

در 9 at se نگاشت شده 9 پینی سرعت زیادی نخواهیم و joystick و keyboard

parallel تقسیم کن به slideهای 8 بیتی و این 8 بیتی را موازی بفرست و برعکس

زمان 1/8 می‌شود که بیشتر است اما پیچیده‌تر است ـ 25 پینی که باید همش وصل بشه
سرعت بالا ها

INT 17H

در برای بهره‌برداری کار کنیم به زبان data در timing های خاص می‌نوشتیم دیگه

Line print parallel Terminal

3/9 18/19

LPT1 → 0 assembly — این بار برای printer استفاده می‌شد و بعد بیشتر

1918 LPT2 → 1 → { 0040:000A = Base address / 0040:000B — حداکثر یکی را داریم

LPT3 → 2 → { 0040:000C / 0040:000D — شماره port وصل اینجاست

LPT4 → 3 → { 0040:000E / 0040:000F — در DOS این LPT را نمی‌خواند

map شده است — در کنار motherboard نوشته می‌شود چند port داریم یا آدرس را از دفترچه این می‌خوانیم ببینیم چه آدرسی map شده

برای استفاده از دستورات IN در OUT باید شماره port را بدانیم — 18 19 در بایت

0040:0008 }
0040:0009 } (1918) LPT1 در BIOS وجود دارد که وقتی run می‌شود شروع می‌کند به load شدن

در بایت برای مشخص کردن شماره port کافی است

مثلاً com port و wired هستند پس شماره port اینجا fix است

Debug.exe قسمتی از حافظه را می‌خواند — D 0040:0008 L8

8 بایت را از اینجا برای من بخوان =>

LPT1	LPT2	LPT3	LPT4
78 03	00 00	00 00	00 00

LPT1 ← 0378H
شماره port

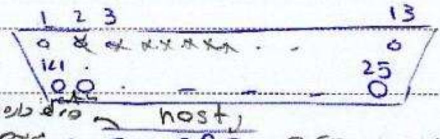
LPT ها وجود ندارد

INIT / Auto Feed

Error
SLCT } اطلاعات به host میدهد ← printer
PE

Subject:
Year. Month. Date. ( )

یک باس 8 بیتی برای پارالل که سرعتش 8 برابر است



host ← 1: STORBE D7 تا D0 در Data خروجی از کامپیوتر است Signal

داده ها [illegible] مرحله صفر نشده تا در [illegible] D0 خوانده نشه

2 D0
3 D1
4 ⋮
9 D7 } Data هست اگر یکی از این سیم‌ها فعال شود یک بیت data نشان میدهند

برای کار کردن آماده است

10 ACK = 0 هست اگر صفر بود یعنی ACK به صورت منفی کار می‌کند ← دریافت تصدیق به data رسیده

11 Busy هست من هنوز Busy هستم data نفرست

12 Out of paper (PE) ← خبر اتمام برای printer کاغذ

13 Select In (SLCT) ← [illegible] کاربرد mode هست خروجی اول کاغذ [illegible] signal هست 1 هست printer is on-line

14 Auto feed ← اجرا کردن [illegible] کند هست یک خط برود برسه اول کاغذ

15 Error = 0 هست ارور printer دارید

16 INIT Initialize printer = 0 (ریست کن) هست دستوری بده ← initiate printer reset

host ← 17 SLC IN Select Input 1 هست Turn printer online register ← D7 تا D0

read only
register 2 ← 11 ، 12 ، 13 ، 15 ، 10

18
⋮
23 } ⏚

اطلاعات [illegible] به host برمی‌گرداند reg 3 [illegible] printer [illegible]
read / write

برای فرستادن data

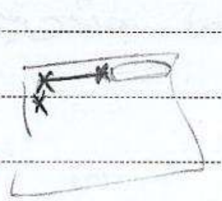
1- صبر کنید تا سیگنال Busy غیرفعال بشود device خارجی آماده است برای دریافت اطلاعات Busy = 0

2- Data

3- سیگنال strobe فعال بشود.

4- فعال شدن ACK ، ACK = 0 ، data برداشته شده می‌توانیم بعدی data را برداریم

به مدت 20 تا 30s صبر کنیم تا ACK = 0 بیاید اگر نیاید چیزی که کار مشکل دارد



P4PCO

Subject:
Year. Month. Date. ( )

INT 17H — تمام کارها را خودش انجام می‌دهد مراحل ۱ تا ۴

منتظر می‌ماند printer وصل شود signal ها را چک می‌کند

function 00 — 01 — 02

ارسال داده — تنظیم اولیه printer — خواندن وضعیت printer

AH ← 00H

DX ← {0, 1, 2, 3} LPT — اینکه با کدام کار کنیم

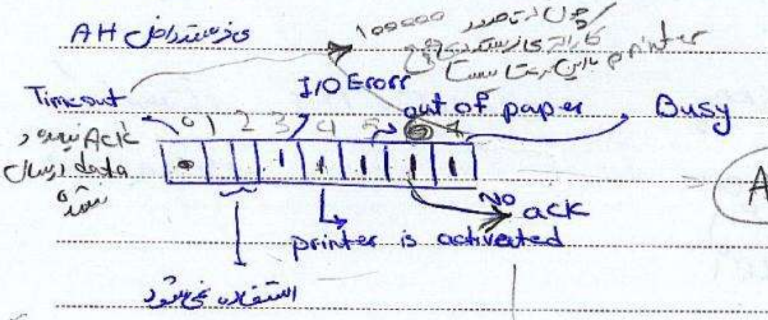
AL ← Data — کد ascii

INT 17H

Data را ارسال می‌کند

status آیا ارسال شد یا نه

می‌ریزد داخل AH



چون از حد مجاز 100000 بار تلاش کرده
printer با این حرکت است
Timeout
I/O Error
out of paper
Busy
0 1 2 3 4 5 6 7
Ack نبوده و data ارسال نشده
No ack
printer is activated
استفاده نمی‌شود

AH / AL → data

AH

اگر printer خاموش است data نباشد یعنی کار اند

و controll می‌فرستیم که خاموش است بگیم خیال شود این ایراد

function 01H — 02H

AH ← 01

DX ← {0, 1, 2, 3}

INT 17H

چاپگر به بالای صفحه می‌رود

آماده برای چاپ می‌شود

AH ← status

AH ← 02

DX ← {0, 1, 2, 3}

INT 17H

AH → status — مثل 01H

در استفاده از دستورات IN, OUT باید شماره port را بدانیم

PAPCO

Subject:
Year. Month. Date. ( )

به‌جای INT می‌توان از دستور IN و OUT استفاده کنیم ولی

قبل از آن باید شماره port را با استفاده از debug.exe پیدا کنیم LP

شماره port که در DX دارد

```
mov AL, <Data>
Out DX, AL
```

0040:000C

به‌جای استفاده از INT از این می‌توانیم درست کنیم

```
mov ES, 40H      [0040:0008]
mov DI, 8H
mov DX, ES:[DI]
```

⇒ برای خواندن اطلاعات درون شماره port

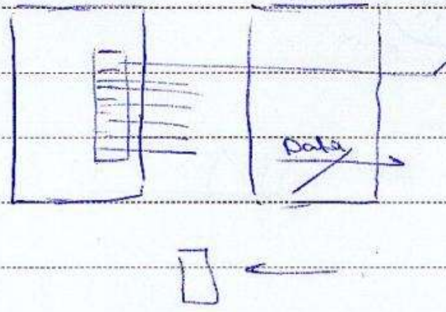
* برنامه‌ای بنویسید که اسم شماره را ببرد port و LPT1 به تعداد ۵ بار ارسال کند.

وقتی از دستورهای IN OUT استفاده می‌کنیم بیت‌های کنترلی را نگاه نمی‌کنیم

حداکثر سرعتی که LPT می‌تواند ارسال کند 12000 KB/s

(USB) سریال است

در حالت‌های ECP تا LPT می‌تواند بیشتر باشد



SAL

CX=5

DEC

AL

BAMDAD

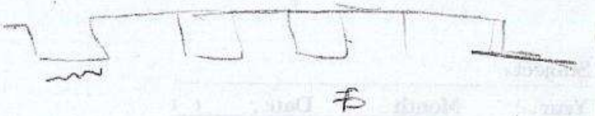
B

B

```
mov AX, 40H
mov ES, AX
mov DI, 8H
mov SI, 0
mov CX, 5   Back:
Back,
while
      mov AL, NAME[SI]
      OUT ES:[DI], AL
      INC SI    cmp SI, 6    JNZ while    1.
Loop Back
```

PAPCO

هزن ~



Serial port

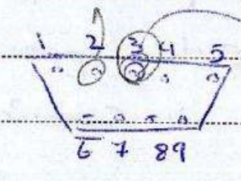
~ serial port . com port

بیت‌های متفاوت را یک یک سیم را جابجا که بتوان تشخیص داده

ارسال { synchrome
AS ~

Serial port interrupt

compert ~ حالت شده

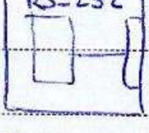


Transmit data

2→1
3→2
4→3

physically ~ جابجا کردیم

~ نیرو ②



com

سری با clock ندارد باشد خوبی هم نیست clock را داشته باشد

در لبه‌های تشخیص مقدار را می‌بیند — جای بیت برهم است — یک start bit - parity bit گاهی

هم Stop bit دارد (8 بیت می‌فرستد) data - 2 یا Stop ، 1 start 1

میزان ارسال بیت در واحد زمان ~ b/s baud rate

الان USB دارد استفاده می‌شود

کاربرد: joystick ، printer ، mouse ، GPS ، external modem

Pin out

استاندارد برای برقراری ارتباط
استاندارد برای ارسال و دریافت

سرور ← PC ارتباط برقرار شده و داده‌ای برای فرستاده شود

pc از مودم می‌پرسد که برای دریافت اطلاعات آماده‌ای

پالس

9 1. DTR (data Terminal Ready)
2 2. recieve data از مودم PC
ارسال داده
3 3 Transmit data ~ DSR، CTS، RTS دقیقا DTR می‌باشند
4 1 DCD (Data carrier detect)
5 . Ground

که کاراکتر ربوط‌ش یک signal دریافت کرده

6. DSR (data Set ready)
7. RTS (request to send)
ready to recieve data
8. CTS (clear to send)
9. Ring Indictor

مودم به pc می‌گوید که یک تماس تلفنی دارد

$^{2}_{3}\!\times^{2}_{3}$ { send مردهم / recieve دهم

سیم رابط متقابل هم وصل می‌شوند یا cross نیز دارد.

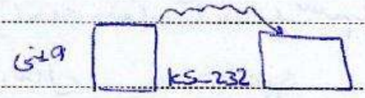
5 ↔ 5

اول برای modem استفاده می‌شد (هم اطلاعات دیجیتالی دارد، هم آنالوگ)

Subject:
Year. Month. Date. ( )

موقعیت x و y که های فرستند با ۰ یا ۰ جدا می‌کند GPRS

به صورت یک رشته‌ی فرستد موقعیت را



① با interrupt bios

② مستقیم در port با in و out کار کنیم راه ①

به شرط اینکه سریال: در حافظه computer می‌گیرید parallel یا serial کجاست در chache چقدر است

ثبت آدرسی

B080:0000 مثل VGA

راه ②

0040:0000 } com1
01

02 } com2
03

04 } com3
05

06 } com4
07

0040:0008 LPT1

مانند: debug.exe

```
_d 0040:0000
```

dump

byte 2 مثل little indian

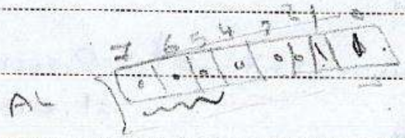
```
_d B800:0000
```

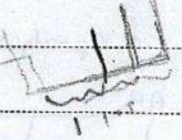
آدرس صفحه مانیتور

Subject:
Year. Month. Date. ( )

BIOS } serial ports
(14H)

func
14H
- 0 → initialize
- 1 → send
- * 2 → receive
- 3 → status





func 0

AH ← 0
DX ← port شماره } 0 ... 3
AL ← (data initialize) ⇝ (3)
INT 14H

AL ← modem status
AH ← serial port status

0 – 1 : { 10 → 7 bit ; 11 → 8 bit — اطلاعات کاراکتر تا 8 بیت باشد حالا کدام است

2 : stop bit (2) { 0 → 1 (stop bit) ; 1 → 2 " " 1.5

3 – 4 : parity { 00 → NO ; 01 → odd ; 10 → NO ; 11 → even

(5–7) : baud rate :

000	110 baud rate
001	150
2	300
3	600
4	1200
5	2400
6	4800
7	9600

7 ← حداکثر نرخ data که می توانیم بفرستیم bious

fun 1

تنظیمات فرستنده و گیرنده (تماس اطلاعات) func0 باید یکی باشد

Send

AH ← 01 حالا مثل است send شده باید نشده باشد
(AL) ← (data)

(DX) ← شماره پورت ... AH 7 = 0 هر چه بر عکس کنیم اینجوری است
همان ترتیب تنظیم زنگ زده باشد

AH → status bit 8 ⇒ $^{AH}_{Bit}$ 7 = 0 شده send
=1 error

P4PCO

Subject:
Year. Month. Date. ( )

func 2

AH ← 02

DX ← port ش 2

INT 14H

AH → status (8bit) Bit 7=0: recieved
=1: error

(AL → data)

func 3

AH ← 03

DX ← port ش 3

INT 14H

AX → status 16 bit
AH = serial port status
AL = modem status

کاملاً مثل status رو بیت به بیت send کرد.

All ← parity error هر بیت‌ها نشان می‌دهند

AH AL data available

7 ... 0 7 ... 0

در برنامه‌ای که data را می‌خواستیم بفرستیم یک حلقه می‌گذاریم که status درست است هر وقت error درست
از loop خارج شود تا زمانی که data را کامل بفرستد

```
| out  DX,AL
| out  250,AL
```

```
outp [0x250] = 'z';
port [$250] := "Z";
```

pos

int ← serial و parallel فقط

dump روی monitor