BIOS 功能经常用于创建 OS（AT 兼容机）

- 我觉得事情变得不再必要，大多数成熟的 32 位模式下的 OS 来了好了，我不知道的事情，当第一次使用或操作系统的引导扇区。

INT（0x10）的关系视频

- 视频模式设置
  - AH = 0X00;
  - AL =模式（较小的屏幕模式为清楚起见，省略）
    - 0x03 的文本颜色 16 的 80x25
    - 0X12：VGA 的图形，640x480x4bit 的颜色，其自身平面内访问
    - 值 0x13：VGA 图形，320x200x8bit 颜色，压缩像素
    - 0x6a：VGA 增强的图形，800x600x4bit 颜色，访问自己的飞机（视频卡不支持。）
  - 返回值：无
- 设置光标形状
  - AH = 0X01;
  - 起始线 CH =;
  - 端线 CL =;
    - 光标一般由两部分组成的 1 CH I <CL
    - 光标由两部分组成，它 CH> CL
    - 光标没有出现，这是 CH == 0x20 的
  - 返回值：无
- 指定的光标位置
  - AH = 0X02;
  - BH = 0;
  - 坐标 DL = X;
  - 坐标 DH = Y;
  - 返回值：无
- 写一个点
  - （鉴于这样的速度，最好是使用更 VRAM 访问自己的，如果这是更好的）
  - AH = 0X0C;
  - AL =颜色代码;
  - 坐标 CX = X;
  - 坐标 DX = Y;
  - 返回值：无
- 字符显示
  - AH = 0X0E;
  - AL =字符代码;
  - BH = 0;
  - BL =颜色代码;

- 返回值：无
- 注：哔哔声，退格键，CR，LF 是公认的控制代码

- 对应的家伙像托盘的数量和颜色代码
  - 这是用在 16 色模式。这并不重要，当设置 256 色模式。
  - AX = 0x1000 的;
  - BL =颜色代码（0-15）
  - BH =调色板代码（0-63）
  - 我们需要与年龄兼容 EGA：注意此功能。
  - 我觉得相当混乱，我认为这是确定 0-15 颜色设置代码，使第 0-15 各调色板代码反正，和我没有混淆的。
  - 一旦出现这种情况，我不必理会那怪异的颜色代码的代码==调色板。
- 调色板设置
  - AX = 0x1010;
  - （0255）;调色板 BX =号
  - DH =红（0∽的 0x3F）
  - CH =绿色;（0∽的 0x3F）
  - CL =蓝（0∽的 0x3F）;
  - 返回值：无
- 显示字符串
  - AH =值 0x13;
  - AL =选项：
    - 为 0x00：BL，光标位置在指定的字符串的颜色不会改变
    - 0X01 指定的字符串在“基本法”的颜色，将光标定位的进步
    - 0x02：将字符串中的颜色属性已被写入，光标的位置不会改变
    - 0x03 的：颜色属性已经写在字符串中，将光标定位在进步，
    - 感觉[代码性质] [颜色代码] [代码性质]像[颜色代码] ...
  - BH = 0;
  - （仅适用于为 0x00，0x01 是一个选项），BL =颜色代码
  - 的长度的字符串 CX =;
  - 坐标 DL = X;
  - 坐标 DH = Y;
  - 字符串的地址 BP =;：ES
  - 返回值：无
- 如何使用模式值 0x13 最简单的屏幕奖励：
  - 这个模式的分辨率是粗糙的，这是很容易处理，因为它是压缩像素。切换屏幕模式，但我必须设置调色板，然后。
  - VRAM 是为 0xa0000 64KB∽0xaffff。严格地说，使 320×200 = 64000，它是 62.5KB。因为它是 1 个字节 1 ==点，请，写 Gashigashi。铅是还 OK。你不需要去画一个点在 BIOS 樽井。！

检索的外设列表; INT（为 0x11）

- 检索的外设列表

- 不必要的调用参数
- 返回：
- 外设列表 AX ==
  - 因为我不认为人们可能忽略的细节。

获取可用的内存的大小，INT（0X12）

- 获取可用的内存的大小
  - 不必要的调用参数
  - 返回：
  - AX =内存大小（以 KB 为单位）

INT（值 0x13）;磁盘关系

- 重新启动系统
  - AH = 0X00;
  - DL =驱动器号（为 0x00∽0x7f 的：FDD，如 0x80∽0xFF 的：HDD）
  - 返回：
  - 没有错误：0 FLAGS.CF ==
  - FLAGS.CF == 1：有一个错误，错误代码 AH
    - （错误代码也可能发生，甚至 HDD FDD）
    - 无效的命令：0X01
    - 0X02：未找到地址标记
    - 0x04 的部门没有发现
    - 0X09：DMA 大于 64KB 的边界，
    - 为 0x10：数据错误
    - 异常控制器：0X20
    - 寻找失败：0X40
    - 0x80 的：超时
    - （这仅发生在 FDD）错误代码
    - 写保护错误写：0x03 的
    - 0X06：未插入光碟
    - 0X08：DMA 溢出
    - （错误代码中可能出现的唯一的 HDD）
    - 故障复位：0X05
    - 为 0x07：无效的参数表
    - 0X0A：Sekutafuragu 是无效的
    - 为 0x11：ECC 数据错误
    - 我还没准备好驱动：把 0xaa
    - 未定义错误：0xbb
    - 写错误：含有 0xCC

- 0XE0：状态错误

- 从磁盘读取，写入到磁盘，验证部门，并寻求
  - （在读）；AH = 0X02
  - （写作）；AH = 0x03 的
  - 在验证时，AH = 0x04 的
  - （SEEK）；AH = 0X0C
  - （可处理连续的扇区），处理 AL =扇区数
  - CH =柱面数和 0XFF;
  - CL =扇区数（位 0-5）| > > 2（＆0x300 到汽缸编号）；
  - 数 DH =头;
  - DL =驱动器号;
  - ES：BX =缓冲区的地址（在时间的验证，这就是不引用寻求）；
  - 返回：
  - 0 FLAGS.CF = =：没有错误，AH = = 0
  - FLAGS.CF = = 1：有一个错误，错误代码（作为复位的功能相同）AH
  - 补充：
    - 注意要处理的扇区数超过指定值时（0X02 为 0x01∽0xFF 的范围内，所以有可能的条件，可以是一个持续不断的过程 - 在 FD 的情况下，也许，不跨过多个轨迹，我想，不应该是超越边界 64KB）
    - 指定的扇区号(如果在 FD 是 0x01∽0x12 中)的范围内为 0x01∽0xff 的
    - 缸数（如果 FD 是 0x00∽0x4f）指定的范围内的量 0x000∽0x3ff 的
    - 为 0x00∽0xFF 的范围内指定头数（FD 的情况下，是 0X00∽0X01）
    - 我不说你应该寻求访问
    - 要访问的硬盘驱动器，你可以注意到，BIOS 不明白的分区，驱动器被视为一个单一的设备在硬件方面。做你自己，如果你需要使用一个分区处理。
    - 我忘了，这是因为在 BIOS（奇异之尘上）的 8 个字节的 0x03 的内容∽0X0A 加载不正确某种原因，当读取引导扇区的 FDD。当你能够控制自己的 FDC 控制 I / O 不会导致这样一个奇怪的事情。
- 驱动器参数
  - AH = 0X08;
  - DL =驱动器号;
  - 返回：
  - AX，BH = = 0 = = 0
  - 光驱类型 BL = =
  - 允许的最大数量＆0xff 的气缸 CH = =
  - > > 2（0x300 到最大的汽缸数，可指定）|（位 5）允许的最大数量的扇区 CL = =
  - 允许的最大磁头数 DH = =
  - 已安装 DL 驱动器的数目= =
  - （这是什么？）地址 DI = = FDD 参数表：ES
  - 补充资料：LBA = =（SEC - 1）+ HED * Max_Sec +缸（Max_Hed 的+ 1）* Max_Sec
    - 如果您已经指定了不规则的几何形状，它不局限于此

- 注：BIOS 不保证返回 0 AX ==
- 硬盘初始化参数表（什么？）
  - AH = 0X09;
  - 返回值：无
- 测试驱动器准备
  - AH = 0X10;
  - DL =驱动器号;
  - 返回：
  - AH ==状态（相同的错误代码？）
- （为什么我寻求缸零？）归一化的驱动器
  - AH =为 0x11;
  - DL =驱动器号;
  - 返回：
  - AH ==状态
- 诊断控制器
  - AH = 0x14 的;
  - 返回：
  - AH ==状态
- 磁盘类型
  - AH = 0x15;
  - DL =驱动器号;
  - 返回：
  - AH ==驱动器类型代码：
    - 为 0x00：在驱动器不存在
    - FDD 检测不到的媒体交流：0X01
    - FDD 可检测到的介质交换：0X02
    - 0X03：HDD
  - 总扇区数 DX ==：CX（仅适用于 HDD）
- 检查交流的 FD
  - AH = 0x16;
  - DL =驱动器号;
  - 返回：
  - 不被取代：0X00 AH ==
  - 被替换为 0x06 AH ==
- 设置磁盘型 FD
  - AH = 0x17;
  - DL =驱动器号;
  - AL =类型代码;
  - 返回值：无
- 物理格式的 FD
  - AH = 0X05;
  - AL = 0X12;
  - CH =气缸;
  - DH =头;

- DL =驱动器号;
- ES：BX =缓冲区的地址;
- 缓冲区是 72 字节，然后在下面的内容（4 字节×18）
  - [的气缸盖 0X01 0X02] [气缸头 0X02 0X02] [汽缸头 0x03 的 0X02] ... [气缸头 0X12 0X02]
- 返回：
- 0 FLAGS.CF ==：没有错误，AH == 0
- FLAGS.CF == 1：有一个错误，错误代码 AH
- 通过此功能：补充，只有一首曲目将被格式化。因此，（也重新建立缓冲每次）所以，在事实上的格式，我也没有格式化的 160 卡车上改变头和气缸。

- 确认存在扩展 INT13H
  - AH = $ 41;
  - DL =驱动器号;
    - 可以用在只有扩展 INT13H 0x80 的∽0xff 的驱动器号
  - BX = 0X55AA;
  - 返回：
  - 我们支持扩展 INT13H：0 FLAGS.CF 的==
  - 我不支持扩展 INT13H：FLAGS.CF == 1
  - （如果支持的话）0xaa55 BX ==：IN13H 扩展的安装
  - BX（如果支持的话）= 0xaa55！IN13H 延长不安装
  - 支持扩展磁盘访问：CL.bit0 == 1（如果支持的话）
  - 不支持扩展磁盘访问：0 CL.bit0 ==（如果支持的话）
- 扩展磁盘读取
  - AH =的 0x42;
  - DL =驱动器号;
  - AL =选项;
    - 正常访问：0 或 1
    - 2：验证访问（以外的设备是不确定的）
    - ：未定义的，否则
  - 以下结构的指针：SI =：DS

抵销	长度	内容
+0 X00	瓦特	（该结构的长度）0x0010 处
+0 X02	瓦特	读的扇区数（1）
+0 X04	瓦特	缓冲区偏移量
+0 X06	瓦特	段选择的缓冲区
+0 X08	8 个字节	64-LBA

  - 返回：
  - 没有错误：0 FLAGS.CF ==
  - 有一个错误：FLAGS.CF == 1

- 扩展磁盘灯
  - AH =×43;
- 验证磁盘扩展
  - AH = 0×44;
- 扩展驱动器控制锁
  - AH = 0X45;
- 媒体弹出的控件扩展
  - AH = 0x46;
- 要求延长
  - AH = 0X47;
- 驱动器的扩展参数
  - AH = 0X48;
  - DL =驱动器号;
  - 指针用来存储以下结构的 SI =：DS

抵销	长度	内容
+0 X00	瓦特	（我会写，可以在呼叫期间接收的最大大小）的整个长度的结构 0X1A
+0 X02	瓦特	旗
+0 X04	ð	物理柱面数
+0 X08	ð	物理磁头数
+0 X0C	ð	的物理扇区的数量
+0 X10	8 个字节	总的扇区数
+0 X18	瓦特	部门长

    - 标志位 0：DMA 边界错误或不发出
    - 信息是否是有效的物理柱面数，物理磁头数，数的物理扇区的扇区长度：标志位 1
    - 无论是可移动驱动器：标志位 2
    - 它是否支持编写与验证：第 3 位标志
  - 返回：
  - 没有错误：0 FLAGS.CF = =
  - 有一个错误：FLAGS.CF = = 1
- 扩展磁盘更换
  - AH = 0X49;
- 启动磁盘仿真可引导的 CD-ROM
  - AH = 0x4a;
- 高端磁盘仿真可引导的 CD-ROM
  - AH = 0x4b;
- 一个可引导的 CD-ROM 的读取状态
  - AH = 0x4b;

- 启动和启动磁盘仿真可引导的 CD-ROM
  - AH = 0x4c;
- 获得引导列表中的可引导的 CD-ROM
  - AH = 0x4d;

INT（0×14），串行端口

- 端口初始化
  - AH = 0X00;
  - AL = BBBPPSCC:
    - 部分 BBB（波特率）：111 = 9600，110 = 4800BPS 101 =为 2400bps，100 = 1200bps 的，011 = 9,600 BPS，010 = 300BPS，001 = 150 基点，000 = 110 个基点
    - 部分 PP（奇偶校验）：00 = 01 =奇数，10 =无 11 =
    - 0 =停止位 1 位，1 个停止位 2 位（停止位）的 S
    - 部分 CC（字符大小）：10 = 11 = 7 位，8 位
  - （例如，0）；DX =端口号
  - 返回值：无
- 提交信
  - AH = 0X01;
  - 要发送到 AL 中的字符=;
  - DX =端口号;
  - 返回:
  - AH == 0，如果没有错误
- 字符接收
  - AH = 0X02;
  - DX =端口号;
  - 返回:
  - 没有错误，如果 AH = 0。
  - AL ==接收到的字符。
- 获得状态
  - AH = 0×03;
  - DX =端口号;
  - 返回:
  - AX =状态;

INT（0x15），其他服务

- 磁带上的马达
  - AH = 0X00;
  - 省略了某些细节，因为我不认为任何人使用此功能。

- OFF 纸盒电机
  - AH = 0X01;
  - 省略了某些细节，因为我不认为任何人使用此功能。
- 读出的数据从磁带盒
  - AH = 0X02;
  - 省略了某些细节，因为我不认为任何人使用此功能。
- 将数据写入磁带
  - AH = 0×03;
  - 省略了某些细节，因为我不认为任何人使用此功能。
- APM 控制
  - AH = 0x53;
  - 欲了解更多信息，APM 到第页上。

INT（0x16）;键盘关系

- 字符阅读
  - AH = 0X00;
  - 返回：
  - AL = ASCII 码，AH ==键盘扫描码
  - 或者，AL == ASCII 的扩展代码 0，AH ==
  - 注：的键缓冲区是空的，没有回来，直到有字符输入
- 检查关键的缓冲状态
  - AH = 0X01;
  - 返回：
  - 有是没有缓冲区中的字符：字符在缓冲区中，1：0 FLAGS.ZF 的==
  - 阅读和输入相同的代码字母 AL，AH。然而，这个角色仍然会保留在缓冲区中。
  - 注：关键缓冲区是空的，很快回来。
- 获取的钥匙锁及移位
  - AH = 0X02;
  - 返回：
  - 状态代码 AL ==：
    - 右移：BIT0
    - 左移：第 1 位
    - 第 2 位：按 Ctrl
    - 第 3 位：ALT
    - 第 4 位：Scroll Lock 键
    - 第 5 位：Num Lock 键
    - 第 6 位：大写锁定
    - 第 7 位：插入模式

INT（0x17）;并行端口

- 输出一个字节。
  - AH = 0X00;
  - 输出数据 AL =;
  - DX =端口号;
  - 返回：
  - AH ==状态：
    - 第 0 位：超时
    - 第 1 位：保留
    - 第 2 位：储备
    - 第 3 位：I / O 错误
    - 第 4 位：打印机被选中，
    - 出纸：第 5 位
    - 第 6 位：承认
    - 第 7 位：忙标志（0：忙，1：Nottobiji）
- 端口初始化
  - AH = 0X01;
  - DX =端口号;
  - 返回：
  - AH ==状态
- 获得状态
  - AH = 0X02;
  - DX =端口号;
  - 返回：
  - AH ==状态

INT（为 0x18）;启动 ROM-BASIC

- 没有参数。我不回来了，也许叫。
- K 表，我要呼吁的家伙未能启动。
- 事实上，兼容的机器在近期的意思是，我没有我 ROM-BASIC，happens've 这样做的，K 表是我已经使用习惯的认识，（我们并不清楚的了解- ·笑）。
  - 我试过在 THINK PAD 770 的续集。这似乎是一个陌生的号码，挂出。

INT（×19）;重新启动

- 没有参数。我不回来了，也许叫。

INT（0X1A）;时间关系

- 读出时钟计数
  - AH = 0X00;
  - 返回：
  - 信号午夜 AL ==
  - CX：DX ==计数
  - （增刊）
    - 计数递增 1 每 54.95 秒，毫秒。
    - 拟做（0X1A）INT AH = 00H 时的最后一次告知是否马修至午夜，午夜的信号，我将返回一个非零值 0，跨不跨。
- 计数时钟设置
  - AH = 0X01;
  - CX：DX =计数;
  - 返回值：无
- 读取实时时钟的时间
  - AH = 0X02;
  - 返回：
  - 在操作过程中，RTC 为 1 :: 0 FLAGS.CF == RTC 停止
  - 当 CH ==（BCD）
  - 分钟 CL =（BCD）
  - 二 DH ==（BCD）
- 实时时钟的时间设置
  - AH = 0×03;
  - 当 CH =（BCD）
  - 会议纪要 CL =（BCD）
  - 第二 DH =（BCD）
  - 时间：DL = 0×00
  - 夏令时：DL = 0X01
  - 返回值：无
- 读取实时时钟日期
  - AH = 0X04;
  - 返回：
  - 在操作过程中，RTC 为 1 :: 0 FLAGS.CF == RTC 停止
  - （地址 0x20 或 0x19）（BCD）CH ==世纪
  - 年 CL =（BCD）
  - 周一 DH ==（BCD）
  - DL ==（BCD）
- 日期设置实时时钟
  - AH = 0X05;
  - （地址 0x20 或 0x19）（BCD）CH =世纪
  - 年 CL =（BCD）
  - 周一 DH =（BCD）
  - 孙 DL =（BCD）
  - 返回值：无

其他典型的矢量（ 2）

- INT（0x1b）;的 Ctrl-Break 处理程序
- INT（为 0x1c）;定时器中断处理程序
- 视频参数的指针; INT（0x1d）
- 的指针磁盘参数 INT（0X1E）
- INT（0X1F）;图形字符表