چند راهنمایی در مورد نوشتن راه انداز سیستم

اول از همه باید بگم که اینکار رو خودتون انجام ندید!!!
حداقل اولین چیزی که میسازید bootloader نباشه!!!به دلایل زیر:

- Bootloader قسمتی از سیستم عامل نیست.
- Bootloader قسمت بسیار کوچکی از سیستم عامل هست.
- نوشتن یک bootloader خوب بسیار پرزحمت و وقت گیر میباشد مانند برنامه نویسی یک سیستم عامل کوچک
- برنامه یک bootloader احتیاج به دانسته هایی دارد که شاید در اول به نظر نمیرسند مانند(unreal mode,A20 gate,BIOS bugs and subtleties)
- دیباگینگ و تست bootloaderها بسیار سخت است.
- بخاطر اینکه باید در دو حالت ۱۶ بیتی و ۳۲ بیتی تقریباً بطور همزمان و هایبرید کارکرد کدهای نوشته شده بسیار سردرگم و پیچیده از آب در می‌آیند.
- اگر هسته سیستم عامل شما از طریق bootloader به مشکل برخورد کند خطایابی آن بسیار مشکل میشود چرا که شما باید هر دو طرف(bootloaderو سیستم عامل) را چک کیند.

اگر شما بر روی داس و یا ویندوز برنامه نویسی میکنید میتوانید از bootloader خود مایکروسافت استفاده کنید(و یا از بوت لودر FreeDOS) ولی اگر بر روی سیستمهای غیر مایکروسافتی برنامه نویسی میکنید GRUB را میتوانید انتخاب کنید.البته GRUB کامل نیست ولی حداقل بهتر کدی هست که شما نوشته اید در واقع اگر loading به مشکلی برخورد کنید میدانید که مشکل از طرف سیستم عامل هست و نه از bootloader و درضمن GRUB بسیاری استاندارها را که شما رعایت نکرده اید رعایت کرده است.

مروری بر bootloaderهای کد باز

GRUB
آدرس سایت:
http://www.gnu.org/software/grub/
چگونگی راه اندازی و استفاده:

http://www.washingdishes.freeuk.com/grubtut.html

grub-how.txt

GRUB سیستم فایل(FAT(DOS/WIN),ext2,Reiserf(Linux) ,FFS(BSD را تشخیص میدهد.GRUB از یک فایل تنظیم در صورت پیدا کردن یکی از فایل سیستمهای ذکر شده استفاده خواهد کرد.فایل تنظیم میتواند تعیین کند که وقتی به سیستم فایل مورد نظر رسید عملیات بارکردن سیستم عامل را انجام دهد و یا صفحه انتخاب سیستم عامل را نشان دهد.

GRUB همچنین دارای یک حالت خط فرمان نیز میباشد که بوسیله آن کنترل بیشتری را به کاربران حرفه ای میدهد و بوسیله آن میتوانید GRUB را تنظیم کرده و یا دوباره آنرا نصب کرده و یا یک سیستم عامل را بصورت دستی بارکنید.

Alexei Frounze's bootloader

آدرس سایت:

http://alexfru.chat.ru/epm.html#bootprog

این bootloader بوسیله خط فرمان لود میشود(مانند داس بوده و از محیط گرافیکی استفاده نمیکند) و به شما اجازه استفاده از FAT12,FAT16 را میدهد در ضمن شما میتوانید دایرکتوری خاصی را انتخاب کرده و هسته سیستم عامل مورد نظر را از آنجا بارکنید در ضمن توانایی بار کردن فایلهای اجرایی داس (.com,.exe) و DJGPP COFF را دارد و با زبان برنامه نویسی Turbo C نوشته شده است و اگر سیستم عامل مورد نظر و یا فایل دیگری از حافظه اولیه ۱ مگابایت + ۶۴ کیلوبایت استفاده نکند به شما اجازه بازگشت به منوی اولیه و انتخاب یک سیستم عامل دیگر را میدهد.

LILO

این bootloader دیگر پشتیبانی نمیشود برای اطلاعات بیشتر میتوانید به آدرس زیر مراجعه نمایید:

.http://metalab.unc.edu/pub/Linux/system/boot/lilo/lilo-t-21.ps.gz

Bootloader داخلی لینوکس

bootsect.s بوت لودر داخلی لینوکس میباشد.

اطلاعاتی در مورد Operation and memory usage

اطلاعات بیشتر : http://www.moses.uklinux.net/patches/lki-1.html

این بوت لودر در kernel 2.6 وجود ندارد.

وظیفه BIOS در راه اندازی سیستم

۱. BIOS سکتور صفر را از دیسک راه انداز بار کرده(CHS=0:0:1) و در آدرس 7C00h قرار میدهد.
۲. BIOS سکتور لود شده را برای بایتهای علامتگذاری چک میکند
   a. 55h at offset 510
   b. 0AAh at offset 511

بسیاری از BIOSها فقط سکتور صفر را میخوانند بدون اینکه به بایتهای علامتگذاری توجه کنند در واقع در این مواقع این وظیفه برگردن MBR هست.

۳. ثبات DL پردازنده به شماره درایو راه انداز تنظیم میشود.
   a. 0h برای فلاپی دیسک (drive A)
   b. 80h برای هارددیسک(drive C)
۴. BIOS به سکتور صفر که تازه لود شده است میرود.

برنامه شما باید ثباتهای زیر را تنظیم کند

- ثبات DS . بعضی از BIOSها این ثبات را با 0 مقداردهی میکنند و بعضی دیگر با 40h . این ثبات باید به یکی از این مقادیر ست شود:
  - (7C00h - BOOT_ORG)/16

مقدار BOOT_ORG مقدار بوت کد شما میباشد که معمولاً همان 7C00h است

- ثباتهای SS و SP (ثباتهای پشته). مقدار اولیه ایی که در این ثباتها قرار میگیرد بستگی به BIOS دارد.
- ثباتهای CSوIP (این ثباتهای را با یک پرش دور دوباره تنظیم کنید).اکثر BIOSها به بوت کد در آدرس 0000:7C00h میروند.اما برخی مانند Compaq Presario 4328 به آدرس 07C0:0000h میروند به دلیل اینکه پرشهای کوتاه و شرطی به ثبات IP وابسطه هستند نیازی به بار کردن دوباره ثباتهایCSوIP نیست اگر کد بوت شما از یک پرش بلند و یا مطلق (JMP) استفاده نمیکند.به هر حال ثبات DS باید حاوی مقدار درست باشد.

اکثر سکتورهای موجود بر روی دیسکها اندازه ای برابر ۵۱۲ بایت دارند.کدی که در BIOS بار میشود باید کمتر از این مقدار یعنی ۵۱۲ باشد خب پس معلوم میشه که باید برنامه را با

اسمبلی نوشت اگر شما نمیتوانید کارهایتان را در یک سکتور انجام دهید باید یک bootloader ۲ مرحله ای که از دو سکتور استفاده میکند بنویسید.

وظیفه MBR در راه اندازی سیستم

سکتور صفر هارددیسک معمولاً حاوی رکورد بوت اصلی (Master Boot Record MBR) میباشد که سه قسمت دارد:

- کدی که باعث لود شدن bootsector از یکی از چهار پارتیشن اصلی میشود.
- Partition level سطح بالا برای هارد دیسک که در افست 446 مستقر هست.این جدول حاوی جدول رکورد چهار پارتیشن میباشد که هرکدام ۱۶ بایت طول دارد.

جدول partition table record:

16-byte partition table record:

offset	size	description
0	1 byte	partition flag byte (0=not bootable, 80h=bootable, or 'active')
1	1	partition start head (H)
2	1	b7:6 = b9:8 of partition start cylinder (C) b5:0 = partition start sector (S)
3	1	b7:0 of partition start C
4	1	OS/filesystem type indicator byte
5	1	partition end H
6	1	b7:6 = b9:8 of partition end C b5:0 = partition end S
7	1	b7:0 of partition end C
8	4 bytes	32-bit LBA first sector of partition
12	4	32-bit LBA number of sectors in partition

در داس و لینوکس ابزاری که جداول پارتیشن ها رو ساخته و یا ویرایش میکند Fdisk نام دارد. در داس دستور مستند نشده fdisk /mbr باعث نصب (دوباره) جدول پارتیشن بر روی هار دیسک میشود.

در فلاپی دیسک MBR وجود ندارد به این علت که در فلاپی قابلیت ایجاد پارتیشن وجود ندارد و حتی میتوانیم هارددیسکی بدون MBR داشته باشیم ولی این روش رایج نیست!

بوت سکتور واقعی

سکتور صفر هارددیسک و یا فلاپی دیسک حاوی بوت سکتور واقعی میباشد.در واقع این هدف اولیه برای پیدا کردن و بارکردن هسته سیستم عامل و یا بار کردن قسمت دوم از یک بوت لودر بزرگتر میباشد.این دو عمل میتواند از طریق یکی از سه راه زیر انجام شود:

۱. بار کردن هسته سیستم عامل و یا قسمت دوم بوت لودر از سکتور مجاور(دیسک، فایل سیستمی ندارد)
۲. بار کردن هسته و یا قسمت دوم بوت لودر از از یک سکتور غیرمجاور(دیسک ممکن است فایل سیستم داشته باشد و یا نداشته باشد)برنامه ای که بوت لودر را بر روی سکتور صفر نصب میکند باید آدرس سکتور غیرمجاور بعدی را نشان دهد.
۳. بار کردن هسته و یا قسمت دوم بوت لودر بوسیله یک فایل سیستم موجود بر روی دیسک.

بریده ای از کد بوت سکتور

خاموش کردن موتور فلاپی درایو:

```
Mov dx,3f2h

Mov al,0

Out dx,al
```

Get memory size with BIOS calls بخاطر مشکلات CMOS این کد ، کدی است که بوسیله آن میتوانید مقدار حافظه PC را بدست آورید.

بدست آوردن مقدار حافظه از CMOS:

```
        ; read extended memory size to AX
        ; won't report more than 63.999 meg (65535/1024) of extended
memory
                 mov al,18h
                out 70h,al
                in al,71h
                mov ah,al
                mov al,17h
                out 70h,al
                in al,71h
                 mov [_ext_mem],ax ; in K
        ; read conventional memory size to AX
                mov al,16h
                out 70h,al
                in al,71h
                mov ah,al
                mov al,15h
                out 70h,al
                in al,71h

        mov [_conv_mem],ax ; in K
```

INT 13h;AH=08h به هر سکتور غیر قابل اعتماد بر روی فلاپی دیسک مقداری را میدهد.فایل سیستم FAT سکتورهای در هر تراک(شیار) را در بلوک پارامتر بایوس (BIOS Prarmeter Block BPB) دخیره میکند.

کد چک کننده پردازنده ۳۲ بیتی در حالت واقعی:

```
pushf
                pushf
                pop bx          ; old FLAGS -> BX
                mov ax,bx
                xor ah,70h      ; try changing b14 (NT)...
                push ax         ; ... or b13:b12 (IOPL)
                popf
                pushf
                pop ax          ; new FLAGS -> AX
        popf
        xor ah,bh               ; 32-bit CPU if we changed NT...
        and ah,70h              ; ...or IOPL
    jne is_32bit_cpu
```

کد چک کننده حالت مجازی ۸۰۸۶(Windows DOS box or EMM386 loader):

```
smsw ax         ; smsw is a 286+ instruction; V86 mode is a 386+
feature
        test al,1       ; look at PE bit of MSW (CR0)
    jne in_v86_mode
```

شما نمیتوانید بطور مستقیم به حالت محافظت شده بروید هنگامی که در حالت مجازی ۸۰۸۶ قرار دارید به خاطر دستورالعملهای LGDTوLIDT و نوشتن ثبات CR0(MSW) که در حالت عملیاتی privilege قرار دارد.اگر سیستم در حالت مجازی ۸۰۸۶ قرار داشته باشد بخاطر مدیریت حافظه مانند EMM386 ابزار VCPI میتواند برای برگشت به حالت محافظت شده صفحه بندی شده(Paged protected mode) مفید واقع شود.

Translated By NETSPC
NETPSC@gmail.com , os@persiasecure.com