به نام خدا

سلام به همگی .

طراحی سیستم عامل قسمت دوم وسایل و پیش نیاز ها

شروع به کار برای ساختن یه سیستم عامل :

قدم اول:
به نظر می رسه که بیشتر مردم دلشون می خواد که یك سیستم عامل بنویسند. اما نمی دونن که از کجا باید شروع کنند و یا اینکه چه زبان یا زبانهایی برای این کار لازم دارند (باید بلد باشند).
خوب , وقتی که من هم می خواستم سیستم عامل خودم رو بسازم مثل شما بودم بنابراین این مقاله ی تمرینی رو براتون نوشتم تا شروع به ساختن یه سیستم عامل برای خودتون بکنید . من این مقاله رو بصورت سوال و جواب نوشتم تا فهمش براتون آسون باشه

۱) سوال : من باید چه زبان های برنامه نویسی رو بلد باشم تا بتونم یه سیستم عامل بسازم ؟

جواب : شما باید زبان اسمبلی ASM رو بلد باشید تا بتونید BootSector خودتون رو بسازید .(بنویسید) (بقیه ی توضیحات تو این زمینه تو سوال بعدیه) و چند تا چیز دیگه . من به شما پیشنهاد می کنم که زبان C/C++ رو هم یاد بگیرید (اقا این علامت + بیخود رفته پشت c . این جلویه C هستش .c++ حالا درست شد . نگید من یه زبان جدید از خودم در آوردم .) چون بیشتر از زبان اسمبلی قابل فهمه . بعد از این شما می تونید هم از زبان اسمبلی و هم از زبان سی تو ساخت سیستم عاملتون استفاده کنین.

۲) سوال :بوت سکتور دیگه چیه ؟

جواب : بوت سکتور (فعال سازی قست اولیه) یه برنامه کوچیکه که تو یه فلاپی دیسک یا هاردیسک نگهداری کرد . این بوت سکتور مسعول پیدا کردن کرنل شما (کرنل قسمت اصلی سیستم عامل شماست که بیشتر کارا رو انجام می ده) بارگذاری اون تویه حافظه و بعدشم اجرای اونه .

۳) سوال : آیا لازمه که من قبل از اینکه یه سیستم عامل بسازم , یه سیستم عامل دیگه رو سیستمم نصب داشته باشم (یعنی یه سیستم عامل دیگه نصب باشه)؟

جواب : آره . بیشتر مردم از ویندوز یا لینوکس برای ساختن یه سیستم عامل استفاده می کنن .

۴) سوال : من باید از چه کمپایلرهایی استفاده کنم ؟

جواب : برای زبان اسمبلی من NASM رو پیشنهاد می کنم . چون اولا هم رایگانه . هم مستندات خوبی داره هم خروجی که داره فرمتهای مختلفی رو پشتیبانی می کنه (یعنی فرمتهای مختلفی رو می تونید ازش بیرون بیارید .) برای اکثر سیستم عامل ها هم در دسترس هست . کد منابعش هم در دسترسه و بیشتر نمونه های اسمبلی بکار رفته برای ساخت یه سیستم عامل به اون نیاز دارن . برای زبان C هم GCC رو پیشنهاد میکنم (البته تحت سیستم عامل های لینوکس و DJGPP و همینطور ویندوز (یعنی اینکه اگه اینا رو نصب کردید) (هر دو تا ی اینا با بقیه سازگارند)

۵) سوال : حالا چجوری من بوت سکتور سیستم عاملم رو Bootable (قابل بالا امدن در اول شروع به کار سیستم عامل) کنم ؟

جواب : شما برای این کار باید خروجی فایل رو رو فرمت Flat binary تنظیم کنید .
یعنی باید فایل شما موقعی که می خواهد از خروجی بیاد بیرون Flat binary باشه .
برای NASM از bin -f استفاده کنین . اگه از C یا C++ استفاده می کنین شما باید از یه ارتباط دهنده (linker) برای این کار استفاده کنید (سوال بعدی رو ببینین)

۶) سوال : این ارتباط دهنده دیگه چیه و چجوری از اون تو ساخت یه سیستم عامل استفاده کنیم ؟
جواب : ارتباط دهنده یا Linker فایلهای خروجی که کمپایلرها بوجود میارند رو میگیره و اونا رو به یك فایل دیگر ارتباط می دهد . برای ساخت یه سیستم عامل , شما بطور معمول می خواهید که فایلهاتون رو با فایلها Flat binary ارتباط بدید , برای LD (ارتباط دهنده ای که بیشتر اوقات با DJGPP و GCC استفاده میشه) از گزینه ی oformat binary-- استفاده کنید . اگر شما از LD استفاده کنید باید از یه اسکریپت ارتباط دهنده برای کنترل بهتر ارتباطها استفاده کنید .پیشنهاد من رو برای ساختن سیستم عاملتون ببینید .

۷) سوال : حالت Real Mode یا مد واقعي چيه ؟

جواب : Real Mode یا حالت واقعي , عنوان لفظيه كه در (X86) (386 486 pnetium) و غيره) وجود داره زمانی که شما برای دفعه ی اول کامپیوترتون رو روشن می کنید . سیستم به این مد مي رود (مد پیش فرض). در واقع در اینجا منظور از مد واقعي همون حالت ۱۶ بيتي هست که سیستم پس از روشن شدن ابتدا در مد واقعي (۱۶ بيتي) هست و پس از انجام دادن كارهاي كه در ادامه مقاله مي خوانيد به مد محافظت شده (۳۲ بيتي) سويچ مي كنيم ويندوز و لينوكس در مد حفاظت شده كار مي كنند (۳۲ بيتي). اما اين حالت واقعي چند تا مشكل اساسی داره:

۱) این حالت از لحاظ امنیتی و محافظتی کمي داره , توانایی که برنامه ها رو از گم به گور شدن و یا باعث هنگ کردن همدیگه میشه باز میداره (این توانایی وجود نداره) مثل محیط داس که شما مي توانيد در هر جاي از حافظه مي خواهيد دادهاتون رو بنويسيد و ممكن شما اشتباهاً در محلي كه سيستم عامل در انجا هست دادهاتون رو بنويسيد و باعث هنگ كردن سيستم عامل بشيد و شايد برنامه شما مشكلي داشته بشه و باعث بشه دادهاي رو در محل هاي حساسي از حافظه مثلا روي سيستم عامل يا بردار وقفه بنويسه و باعث فروپاشي (كراش) سيستم عامل بشه يعني از برنامه ها براي دسترسي به حافظه ديگر برنامه ها در حال اجرا محفاظت نميشه

۲) شما فقط مي توانيد از يك برنامه در آن واحد استفاده (اجرا) كنيد حتي اگر شما قابليت چند ريسماني رو براي سيستم عاملتون در مد واقعي فراهم كنيد بازهم به دليلي كه در قسمت قبلي گفتم سيستمتون پايدار نيست و اشكال در يك برنامه باعث كراش شدن كل سيستم ميشه يعني شما اگر براي سيستم عاملتون در مد واقعي قابليت چند ريسماني فراهم تنها امكان كراش شدن بيشتر سيستم عاملتون رو فراهم كرديد پس شما نمي تونيد در اين مد چند ريسماني روبكار بگيريد و شما محدود به اجراي يك برنامه در هر لحظه هستيد

۳) شما تنها به يك (۱) مگابایت فضا برای ادرس دهی محدود هستید.
۴) شما حتما باید از سگمنت ها (Segment) برای دسترسی به حافظه استفاده کنین

۸) سوال: حالت Protedcted Mode یا مد محافظت شده چیه؟
جواب :حالت محافظت شده (به PMode هم معروف هست) , یه مد پیشرفته تر از از حالت واقعي يا Real Mode هست

حالت محافظت شده نسبت به حالت واقعي چندين مزيت داره :

۱) حالت محافظت شده , حالتي هست كه برنامه ها از دسترسي به حافظه همديگر حفاظت مي شوند. قابلیت و توانایی که باعث میشه , برنامه گم و گور نشن و یا باعث اختلال در کار هم نشوند . (اسم این حالت هم به همین دلیل Protedcted هست.)

۲) این امکان هست که شما چند برنامه همزمان در حال اجرا داشته باشید (مثل لینوکس یا ویندوز)

۳) شما به ۴ گیگابایت فضا برای ادرس دهی دسترسي داريد , آره كف نكنيد ۴ گيگابايت فضا براي ادرس دهي داريد (زمانی که درگاه A20 فعال شده باشه درباره A20 سوال نکنید . سوال بعدی رو ببینید .)

۴) شما میتونید از صحفه ها براي دسترسي به حافظه استفاده کنید بر خلاف مد واقعي که سگمنتها براي دسترسي به استفاده مي شوند در مد حفاظت شده از صحفه ها براي دسترسي به حافظه استفاده مي شود با استفاده از صحفه بندي و چند قابليت ديگر ريز پردازنده امکان پياده سازي حافظه مجازي براي ما فراهم مي شود و ما مي توانيم به حافظه نا محدودي دسترسي داشته باشيم (صحبت در مورد صحفه بندي مفصل هست و نميشه در اين مقاله بهش پرداخت و خودش يك مقاله ديگري طلب مي کنه . براي اطلاعات بيشتر در مورد صحفه بندي مي توانيد کتابهاي که در مورد سيستم عامل هست رو مطالعه کنيد)

۵) اگر تا الان نگرفتين بايد بگم که سيستم عامل شما بايد در حالت حفاظت شده اجرا بشه . مگر اينکه شما دليل خاصی داشته باشيد که بخواين از حالت مزخرف مد واقعي استفاده کنيد .(پس ما در حال عادی از حالت حفاظت شده برای اجرای سيستم عامل استفاده می کنيم.)

۹) سوال : راستی اون درگاه A20 که گفتی , چيه ؟

جواب:ريز پردازندها ۸۰۸۶ (پردازنده ای که قبل از ۲۸۶،۳۸۶،۴۸۶و پنتيوم و غيره ... اومدن) تنها ميتونستن به يک مگابايت فضا برای آدرس دهی دسترسی داشته باشند . برای اون موقع گستره مقدار حافظه ۲۰ تا ادرس ۱ بيتی(نه بايت !) از A0 تا A19 سودمند و مفيد بود . بعدش اينتل ۸۰۲۸۶ رو اختراع کرد و ساخت (که بعدا فقط بهش ۲۸۶ ميگفتن) ۸۰۲۸۶ می تونست تا ۱۶ مگابايت فضا برای ادرس دهی دسترسی پيدا کنه . ولی ۸۰۲۸۶ بايد با ۸۰۸۶ سازگار مي شد . (برنامه هايی بودند که برای ۸۰۸۶ ساخته شده بودند که حالا بايد روي ۸۰۲۸۶ اجرا بشوند .) حالا خر بيارو باقالی بار کن . حالا راه حل برای اينکه ۸۰۲۸۶ قابليت دسترسی به فضای مورد نياز برای ادرس دهيش بيشتر شه چيه ؟ در عين حال بعضی برنامه ها بر روی اين واقعيت که ادرسها دور وره يه چيزی حدود ۱ مگابايت جمع اوری و ساخته شده شدن, استوار بود . بچه ها , بياين اينو طوری که ۸۰۸۶ ميديد ما هم ببينيم , شما ادرس ۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱ (۲۰تا ۱) رو در يک سيستم دوتايی (Dual asystem) داريد و به اون يه يک (۱) اضافه ميکنيد . فکر ميکنيد چه اتفاقی می افته , نتيجه صفره . چونکه شما بيست و يکمين بيت رو نداريد (A20!) ولی در واقع ۸۰۲۸۶ بيت های بيشتری داشت . برای حل اين مشکل اينتل ۸۰۲۸۶ رو ساخت , درگاه A20 رو غير فعال کرد و گستره ی ادرس دهی رو به ۲۰ بيت (A20) رسوند .(محدود کرد). خوب ,پس برای دسترسی به بيشتر از يک مگابايت حافظه , ما بايد درگاه ۲۰ يا A20 رو فعال کنيم .

۱۰) سوال : انتخاب گر يا Selector چيه ؟
يه انتخاب گر (بعضی وقت ها , بهش شناسه يا توصيف گر هم می گويند .) معنيه يه بخشه . يعنی اينکه کار و وظيه ای که براش از قبل تعيين شده ,رو مشخص می کنه) که اطلاعاتی از و ظايف و کار های بخش پايه رو در خودش داره . مثل اساس بخش (Segment base) (اين شروع کننده است) و حد بخش (اين طولشه) . هر انتخابگر ۶۴ بيتی هست . اطلاعات بيشتر از اين دوتا رو تو قسمت GDT و LDT بخونيد .

۱۱) سوال : GDT چيه ؟
GDT يه ميزه (يه بخش) که حاوی انتخابگره . اساسا GDT ادرس اوليه رو برای ادرس دهی تمامی انتخاب گر ها بو جود می اره . (صحبت در مورد GDT مفصل هست و خودش يك مقاله جداگانه طلب مي کند)

۱۱) سوال : Interrupt يا وقفه چيه؟
جواب : وقفه يا Interrupt يه علامته که باعث ميشه cpu از انجام عمليا‌تی که در حال حاضر در حال انجامشه دست بکشه . اين به cpu ميگه که يه عمليات مهم رسيده , برای مثال يك قطعه سخت افزاري مثل موس با هر حرکت يا فشرده شدنش (کليك) يك پيغام به CPU مي فرسته و باعث ميشه که CPU برنامه اي که در حال اجرا داره رو به طور موقت تا زمان پاسخوگوي به وقفه کنسل کنه و سپس شماره وقفه داده شده رو مي گيره (هر سخت افزار خارجي يك شماره وقفه خاص داره) و متناسب با اون شماره از جدول بردار وقفه که مشخص هست (مثلا در داس از0x0000 تا 0x03FF هست) ادرس روالي (تابع يا زير برنامه) که بايد با فعال شدن اين وقفه اجرا بشه رو مي گيره و بعد محل روالي که در حال اجرا کردنش بود رو به علاوه تمام رجيسترها رو در پشته ذخيره مي کنه و CPU به اين محل (تابع وقفه) پرش مي کنه و بعد از اجرا شدن اين تابع دوباره آدرس برنامه اي که در حال اجراش بود رو از پشته بر مي داره به علاوه تمام رجيستر ها و دوباره کارش رو ادامه مي دهد (انجام برنامه اي که اول در حال اجراش بود) همونطور که می دونيد وقفه ها. تنظيمات مخصوص به خودشونو

دارن . برای مثال , اونا می تونن از Ring 3 به Ring 0 تغییر پیدا کنن و مورد استفاده قرار بگیرن بیشتر خود سیستم عامل ها اینکار رو اننجام مي دهند (عوض کردن Ring ها) .اونا می تونن در یه سیستم عامل به عنوان راهی برای انجام مکالمات سیستم استفاده بشن .مکالمات سیستمی لینوکس هم بر همین مبناست . (در واقع وقفه هستند که تمام کارها سیستم رو انجام مي دهند و بي نهايت مهم هستند وقفه ها به سرعت اجرا مي شوند و دقت کنيد همين حالا که داريد اين متن رو مطالعه مي کنيد در هر ثانيه بيش از ۱۰۰ ها وقفه در سيستم شما اتفاق مي افتد !)

۱۳) سوال : PIC چیه ؟
جواب : مخفف کنترل گر وقفه قابل برنامه ریزی است , يك وسيله ای هست که در خواست های سخت افزار ها برای وقفه (خاتمه ی یک عملیات و شروع عملیات دیگر) رو جمع اوری میکند. و به Cpu علامت می ده که یه وقفه اتفاق افتاد . وقتی که این اتفاق می افته , هم PIC و Cpu هر دو با هم ارتباط بر قرار می کنن و وقفه گر رو برای اجرای درخواست جدید اجرا می کنن .

۱۴) سوال : حالا این PIT یعنی چی ؟
PIT مخفف زمانسنج وقفه قابل برنامه ریزی هست . یه زمانسنج سخت افزاری قابل برنامه ریزی (شما می تونید سرعتش رو تغییر بدهید) هستش که وقفه رو از بین می بره زمانی که کارش تموم می شه اي زمان سنج بر روي خط صفر وقفه مي دهد (IRQ O) اين وقفه اصلي هست که براي زمان بندي Task ها از استفاده مي شود (با اين وقفه چند ريسماني پياده سازي مي شود)

۱۵) سوال : بر چه اساس و ترتیبی من باید سیستم عاملمو بسازم ؟
جواب :خوب شما می تونید , تقریبا از هر جایی شروع کنید , ولی مطمعنا وقتی کارتون تموم شد یه دو جین باگ و سوراخ سمبه پیدا می کنید . من پیشنهاد می کنم که شما اول یه بوت سکتور بسازین , بعدش یه کرنل پایه بسازید و بعد هم بقیه رو به کرنل اضافه کنین .

۱۶)سوال : بازم یه سوال دیگه :قدمهای اساسی و اولیه برای وارد شدن به حالت محافظت شده در موقع بوت چیه ؟

جواب:
۱) وقفه ها رو غیر فعال کنید تا چیزی گم و گور نشه تا در موقع وارد شدن به مد محافظت شده وقفه اي اجرا نشه چون اگر در موقع وارد شدن به مد محافظت شده وقفه اي جرا بشه به احتمال زياد سيستم هنگ (کراش) مي کند
۲) GDT رو نصب کنید (با دستور Lgdt در این مورد در مقالات بعدي صحبت خواهد شد) .
۳) A20 رو فعال کنید .
۴) اولین بیت از رجیستر CR0 رو ۱ کنید.
۵) مقدار مناسب رو در سگمت هاي ES و DS و SS و FS و GS قرار دهيد (در اين مورد در مقالات بعدي توضيح داده خواهد شد)
۶) رجیستر و سگمنت CS و EIP رو با مقدار های درست , بوسیله ی یک پرش بلند, بارگذاری کنید . (در اين مورد در مقالات بعدي توضيح داده خواهد شد)
۷) حالا شما تو حالت محافظت شده هستید !

ThanX t0 th3 Master f0r Translati0n
ThanX t0 th3 _LOVE_CODER_ f0r advic3
C0nv3rt3d by netspc