S.A.H.A

دستورات اسمبلی(قسمت اول)

Registers :

ثبات بطور ساده محلی برای ذخیره اطلاعات است. چند ثبات مهم که بیشتر مورد استفاده ما قرار میگیره عبارتند از :

AX , BX , CX , DX ;16 bit

EAX , EBX , ECX , EDX ;32 bit

البته در محیط ویندوز که یک محیط ۳۲ بیتی هستش در اول اسم همه این ثباتها یک E قرار میگیره

مثلا : EAX

FLAGS :

فلگها هم شبیه ثباتها هستند بجز اینکه فقط قادر به گرفتن مقادیر صفر و یک یا غلط و درست هستند.

فلگها توسط دستورهایی مانند CMP , JNE و . . . تنظیم میشن و از اونها برای چک کردن خروجی ها مانند خروجی یک عمل مقایسه ، استفاده میشه.

Stack :

استک یا به فارسی همون پشته محلی برای ذخیره سازی ترتیبی داده هاست برای استفاده مجدد.

هر برنامه قبل از فراخوانی تابع باید تمام مقادیر لازم برای تابع رو درون استک قرار بده.

استک رو مثل چند بشقاب روی هم فرض کنید که در اونها اولین بشقابی که قرار میگیره در حقیقت آخرین بشقابیه که برداشته خواهد شد.

این مساله خیلی مهمه که حواستون باشه اولین عضوی که به پشته فرستاده میشه ، آخرین عضویه که برداشته خواهد شد و این یعنی اینکه پارامترهای یک تابع بصورت برعکس از آخر به اول در پشته قرار میگیرن.

از دستور Push برای قرار دادن یک مقدار در پشته بصورت زیر استفاده میشه :

PUSH EAX

این دستور مقدار ثبات EAX رو در پشته قرار میده.

از دستور Pop هم همانند دستور Push استفاده میشه اما برای برداشتن یک مقدار از داخل استک.

دستور AND

شکل استفاده : AND dest,src

هدف : یک عمل AND منطقی بین dest و src انجام میده و نتیجه رو توی dest ذخیره میکنه.

مثال : AND BX,03h

در این مثال مقدار داخل ثبات BX با مقدار هگز ۰۳ ، AND میشه و جواب محاسبه توی ثبات BX ذخیره میشه.

دستور CALL

شکل استفاده : CALL address

هدف : تابعی رو که در آدرس "address" قرار داره اجرا میکنه.

مثال : CALL 10284320

این دستور تابعی رو که در آدرس ۱۰۲۸۴۳۲۰ در حافظه قرار داره اجرا میکنه.

دستور CMP

شکل استفاده : CMP dest,src

هدف : مقایسه دو پارامتر با یکدیگر و ذخیره پاسخ در فلگ ها

مثال : CMP AX,03h

این دستور یکی از دستورات مهم تو کار کرکه. از این دستور برای مقایسه دو مقدار در یک برنامه استفاده میشه مثلا مقایسه دو شماره سریال !

جواب مقایسه (درست یا نادرست بودن مقایسه) در فلگ ها (معمولا فلگ صفر ، فلگ صفر یکی از فلگ های موجود بین فلگ هاست که در انجام اعمالی مثل مقایسه مورد استفاده قرار میگیره)ذخیره میشه.

دستور INT

شکل استفاده : INT int_num

هدف : فراخوانی یک تابع که معمولا در BIOS ذخیره میشود

مثال : INT 10h

بطور کل شما در کرکینگ برنامه های تحت ویندوز تقریبا هیچ وقت با این دستور برخورد نمیکنید ولی دونستن این دستور مفید خواهد بود.

دستور JMP

شکل استفاده : JMP address

هدف : پرش به قسمتی از کد برنامه

مثال : JMP 00402011

این دستور یک پرش غیر شرطی هست که به محل مورد نظر در کد پرش میکنه.

دستورهای مختلفی در رابطه با پرشهای شرطی وجود داره که مهمترین های اونها برای ما به شرح زیر هستند :

JZ که در صورتی که فلگ صفر ، ست شده باشه (یعنی دارای مقدار یک باشه) پرشی به محل مشخص شده در جلوی اون انجام میده. مثال : JZ 00402011

JNZ که در صورتی که فلگ صفر ، ست نشده باشه (یعنی دارای مقدار صفر باشه) پرشی به محل مشخص شده در جلوی اون انجام میده. مثال : JNZ 00402011

JE که همون کار JZ رو انجام میده .

JNE که همون کار JNZ رو انجام میده.

این دستورات پرش شرطی معمولا بعد از انجام یک عمل مقایسه می آیند.

مثال :

BadSerial , CMP RealSerial

GoAwayBadCracker JNE

البته این یک مثاله و در واقعیت از این کلمات استفاده نمیشه !

دستور MOV

شکل استفاده : MOV dest,src

هدف : مقدار src رو به dest کپی میکنه

مثال : MOV AX,DX

شما وقتی که درون یک کد جلو برید با تعداد زیادی از این دستورات خواهید دید.

در حقیقت کار این دستور بصورت زیره :

AX = DX

این دستور چند مشابه هم داره مثل MOVSX که تقریبا همون کار رو انجام میدن.

دستور OR

شکل استفاده : OR dest,src

هدف : یک عمل OR منطقی انجام میده و نتیجه رو در dest ذخیره میکنه.

مثال : OR DX,AX

دستور RET

شکل استفاده : RET

هدف : برای بازگشت از یک تابع به برنامه فراخواننده تابع استفاده میشود.

مثال : RET

این دستور معمولا در آخر یک تابع دیده میشه و باعث میشه که کنترل برنامه به کدی که تابع از اونجا فراخوانی شده برگرده.

دستور INC

شکل استفاده : INC dest,src

هدف : محنوی ثبات را یک واحد افزایش میدهد

مثال : inc AX

دستور DEC

شکل استفاده : DEC dest,src

هدف : محنوی ثبات را یک واحد کاهش میدهد.

مثال : dec AX

hooooooooooooooooooooooooooooooootCrack!

mahdiasadi@yahoo.com

mahdilive@yahoo.com

www.sahagroup.org