* عملکرد دستور cbw را با ذکر مثال توضیح دهید.

ج

در برنامه‌نویسی اسمبلی مخصوصاً هنگام کار با پردازنده‌های اینتل، دستورات نقش مهمی در تعامل با سخت‌افزار را ایفا می‌نمایند که یکی از دستورات، دستور cbw است.

دستور cbw به طور خاص در معماری X86 استفاده می‌شود و عملکرد آن به این صورت است:

مقدار موجود در ثبات AL را می‌گیرد و آن را به یک مقدار 16 بیتی در ثبات AX گسترش می‌دهد.

اگر بیت بالاترین (یا همان بیت علامت) در AL برابر 1 باشد (یعنی عدد منفی است) بالاترین بایت (AH) با مقدار FFh پر می‌شود یعنی: 11111111.

اگر بیت علامت 0 باشد یعنی عدد مثبت است، بالاترین بایت (AH) با 00h پر می‌شود.

به عبارت دیگر، این دستور علامت یک مقدار 8 بیتی در AL را در قسمت بالایی AX در (AH) گسترش می‌دهد.

مثال:

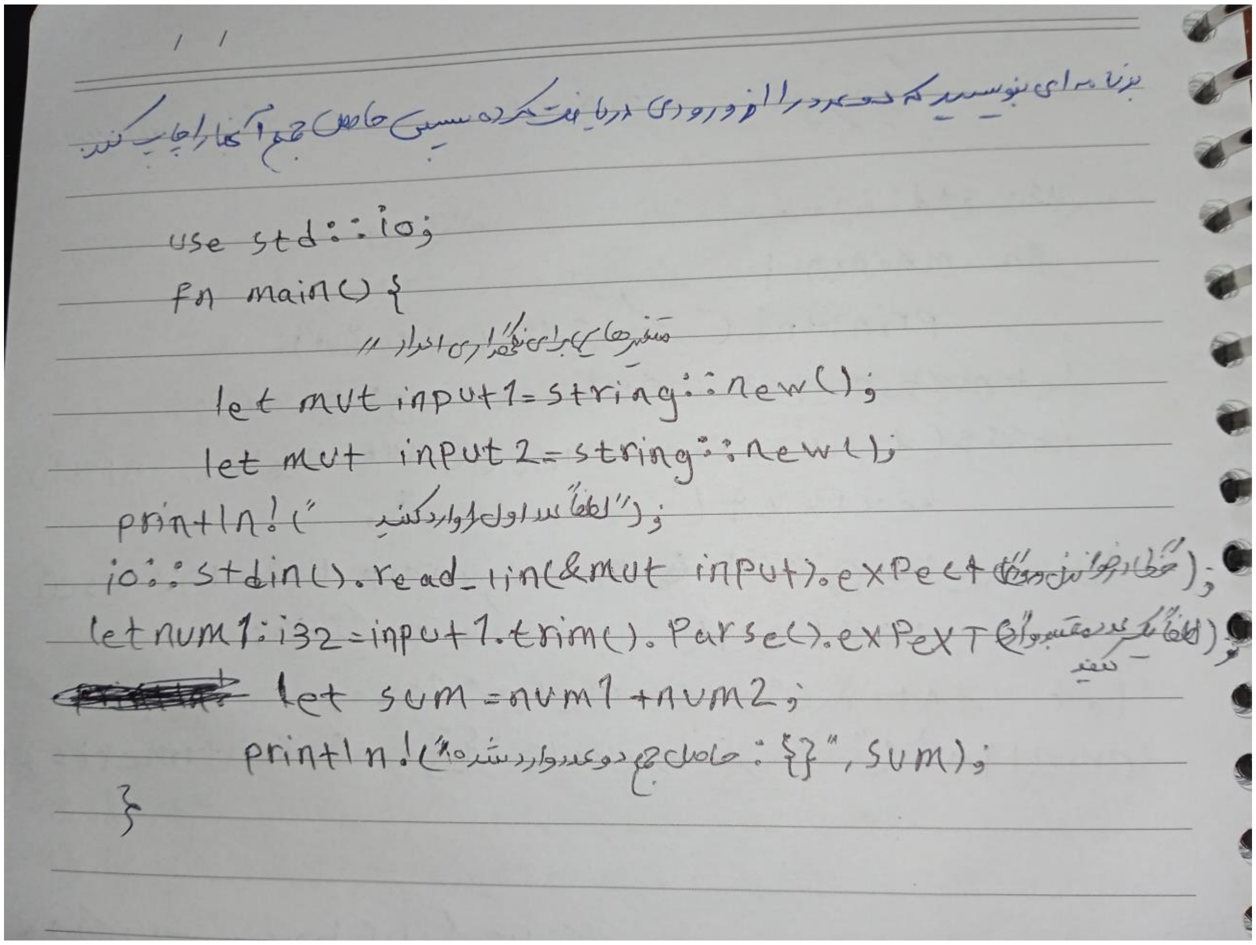
فرض کنید در AL مقدار -10 (یعنی F6H در شکل هگزادسیمال) موجود است. وقتی دستور cbw اجرا می‌شود:

```
MOV AL, -10        ; AL = 0xF6
cbw                ; AX = 0xFFF6
```

در این مثال چون AL (F6H) عددی منفی است، مقدار FFh به AH منتقل می‌شود تا نشان دهد که AX همچنان منفی است و عدد نهایی در AX برابر با FFF6H خواهد بود.

۱- برنامه‌ای بنویسید که دو عدد را از ورودی دریافت کرده سپس حاصل ضرب آن‌ها را چاپ کند.

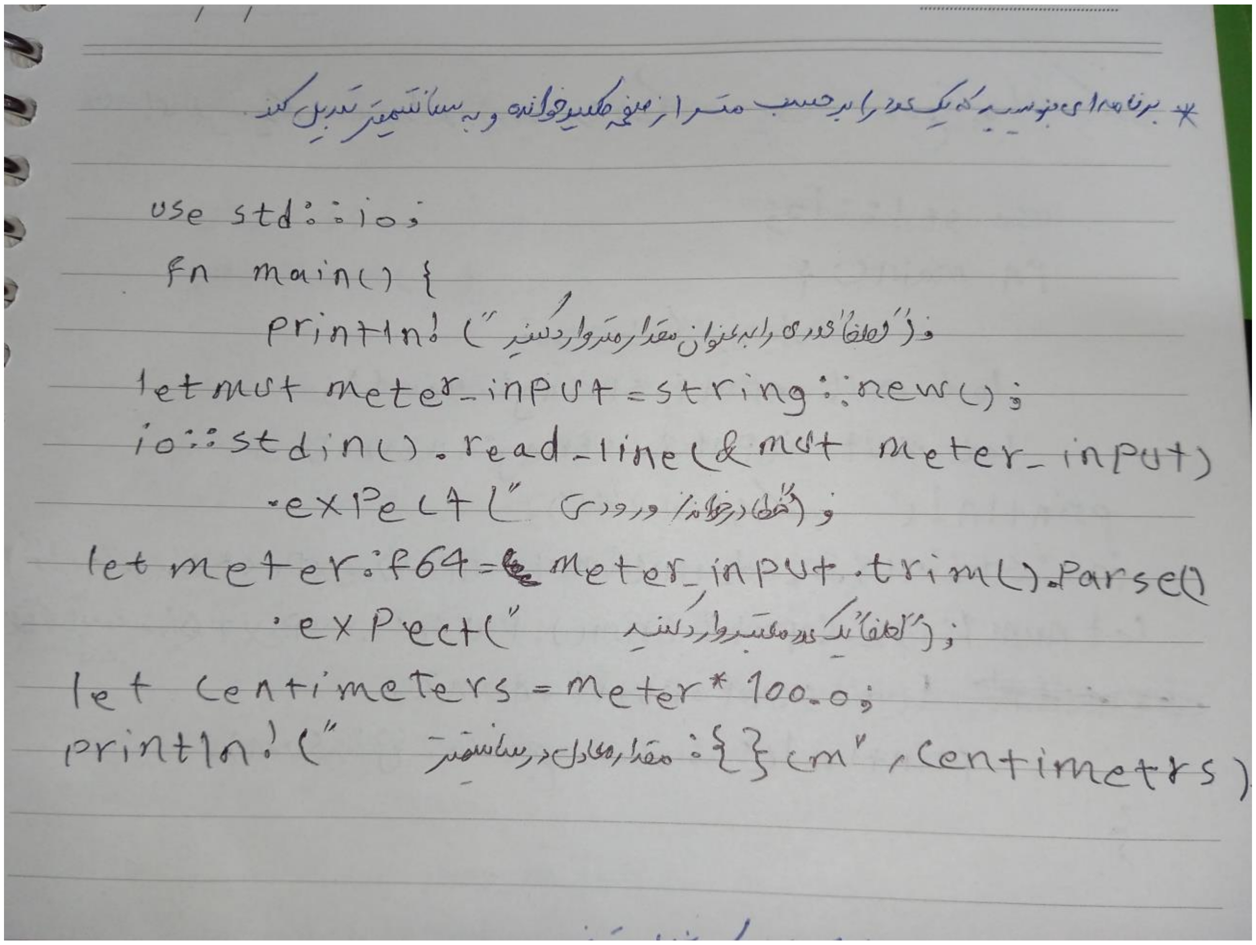
```
Use std::io;
fn main() {
    // تعریف متغیر برای ورودی اول
    let mut input1 = string::New();
    println!("لطفاً عدد اول را وارد کنید");
    io::stdin()
        .read_line(&mut input1)
        .expect("خطا در خواندن ورودی");
    let num1:i32=input1.trim().Parse().expect("
    "لطفاً یک عدد صحیح وارد کنید");

    // تعریف متغیر برای ورودی دوم
    let mut input2 =string::New();
    println!("لطفاً عدد دوم را وارد کنید");
    io::stdin()
        .read_line(&mut input2)
        .expect("خطا در خواندن ورودی");
    let num2 :i32=input2.trim().Parse().expect("
    "لطفاً یک عدد صحیح وارد کنید");

    // محاسبه و چاپ حاصل ضرب
    let result=num1* num2;
    Println!("حاصل ضرب {} و {} برابر است با :{}",num1,num2,
                                      result);
```



/ /

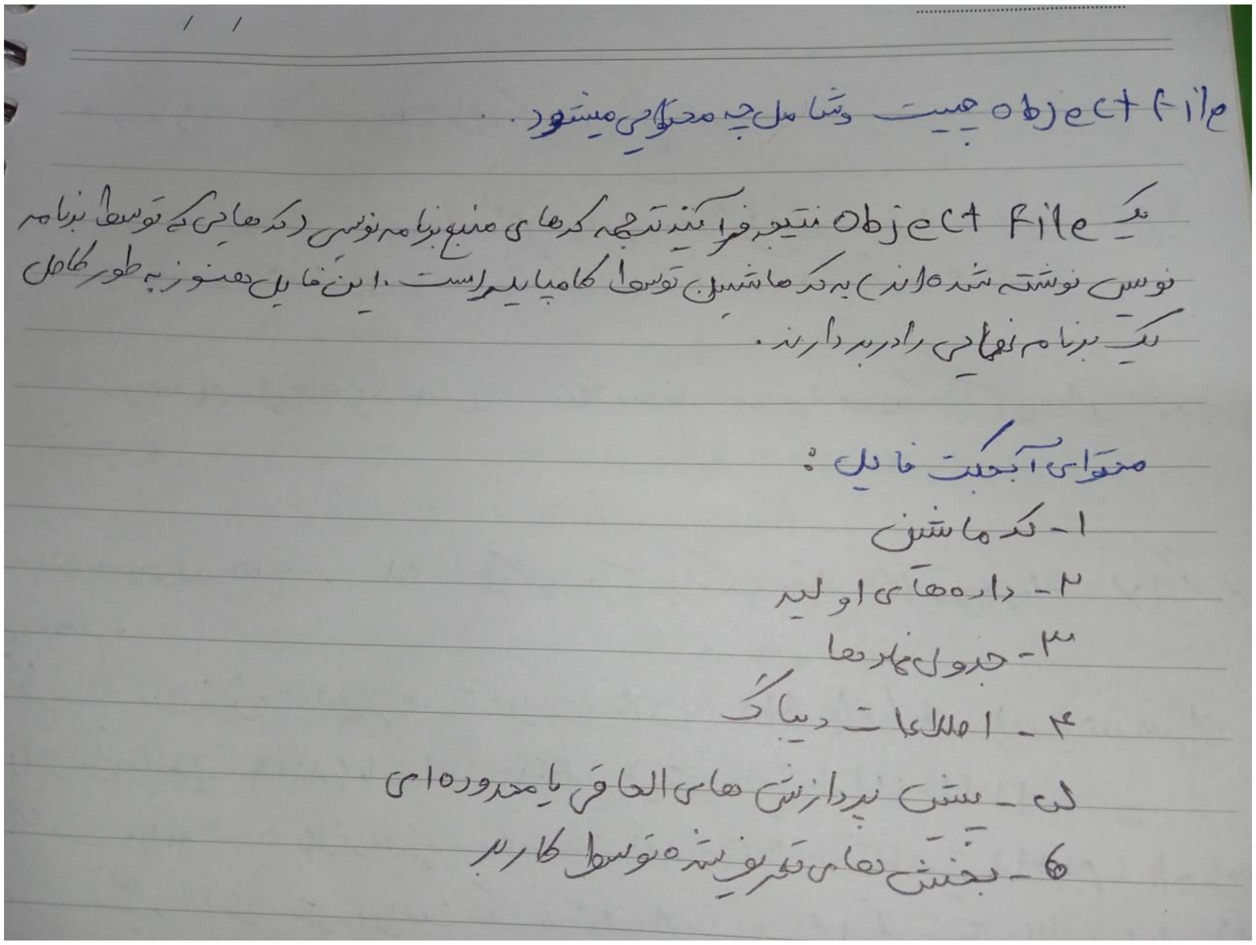
برنامه‌ای بنویسید که دو عدد را از ورودی دریافت کرده سپس حاصل جمع آنها را چاپ کند.

```
use std::io;
fn main(){
            // متغیرهای برای نگهداری اعداد //
    let mut input1=string::new();
    let mut input2=string::new();
println!("لطفا عدد اول را وارد کنید");
io::stdin().read_lin(&mut input).expect("خطا در خواندن ورودی");
let num1:i32=input1.trim().Parse().expexT("لطفا یک عدد معتبر وارد کنید");
~~[illegible]~~ let sum=num1+num2;
        println!("حاصل جمع دو عدد وارد شده: {}", sum);
}
```



* برنامه‌ای بنویسید که یک عدد را برحسب متر از صفحه کلید بخواند و به سانتیمتر تبدیل کند

```
use std::io;
 fn main() {
        println!("لطفاً عددی را به عنوان مقدار متر وارد کنید");
 let mut meter_input = string::new();
 io::stdin().read_line(&mut meter_input)
        .expect("خطا در خواندن ورودی");
let meter:f64 = meter_input.trim().parse()
        .expect("لطفاً یک عدد معتبر وارد کنید");
let centimeters = meter * 100.0;
println!("مقدار معادل در سانتیمتر: {} cm", centimetrs)
```



/ /

object file چیست و شامل چه محتوایی میشود.

یک object file نتیجه فرآیند ترجمه کدهای منبع برنامه نویسی (کدهایی که توسط برنامه نویس نوشته شده اند) به کد ماشین توسط کامپایلر است. این فایل هنوز به طور کامل یک برنامه نهایی را در بر دارند.

محتوای آبجکت فایل:

۱- کد ماشین

۲- داده های اولیه

۳- جدول نمادها

۴- اطلاعات دیباگ

۵- پیش پردازش های الحاقی یا محدوده ای

۶- بخش های تعریف شده توسط کاربر

/ /

تبدیل اعداد از مبناهای ۲، ۸ و ۱۶ به ۱۰ را با ذکر مثال بیان کنید.

۲ به ۱۰

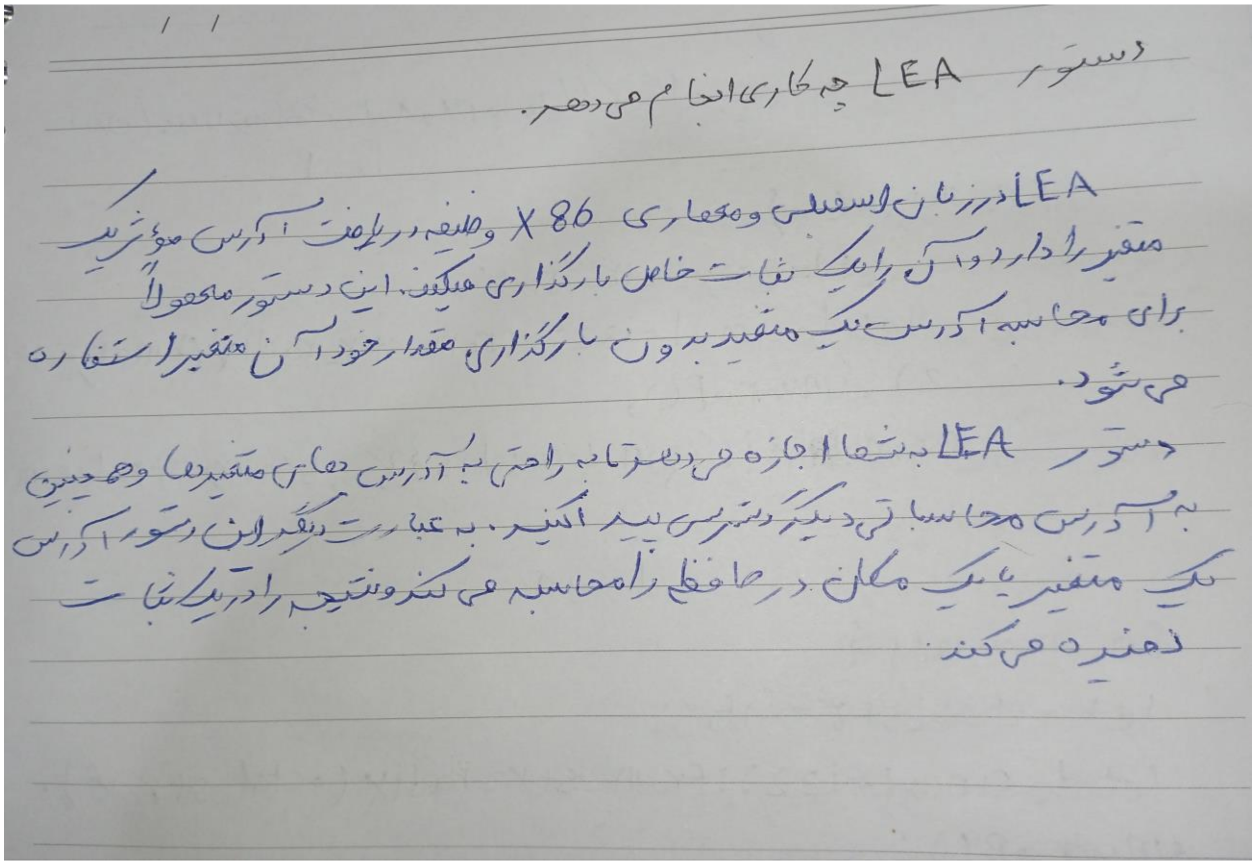
```
fn main(){
    let binary_str="1101";
    let decimal=i32::from_str_radix(binary_str,
    2),unwraP();
    println!("{} in decimal is {}' ,binary_str,decimal);
```

۸ به ۱۰

```
fn main(){
let octal_str="15";
let decimal=i32::from_str_radix(octal_str, 8).
unwrap();
 Println!("{} in decimal is {}',octal_str,decimal);
```

۱۶ به ۱۰

```
fn  main(){
let hex_str="D";
let decimal=i32::from_str radix(hex,16).
                                      unwraP();
Println!("{} in decimal is{}",hex_str,decimal);
}
```

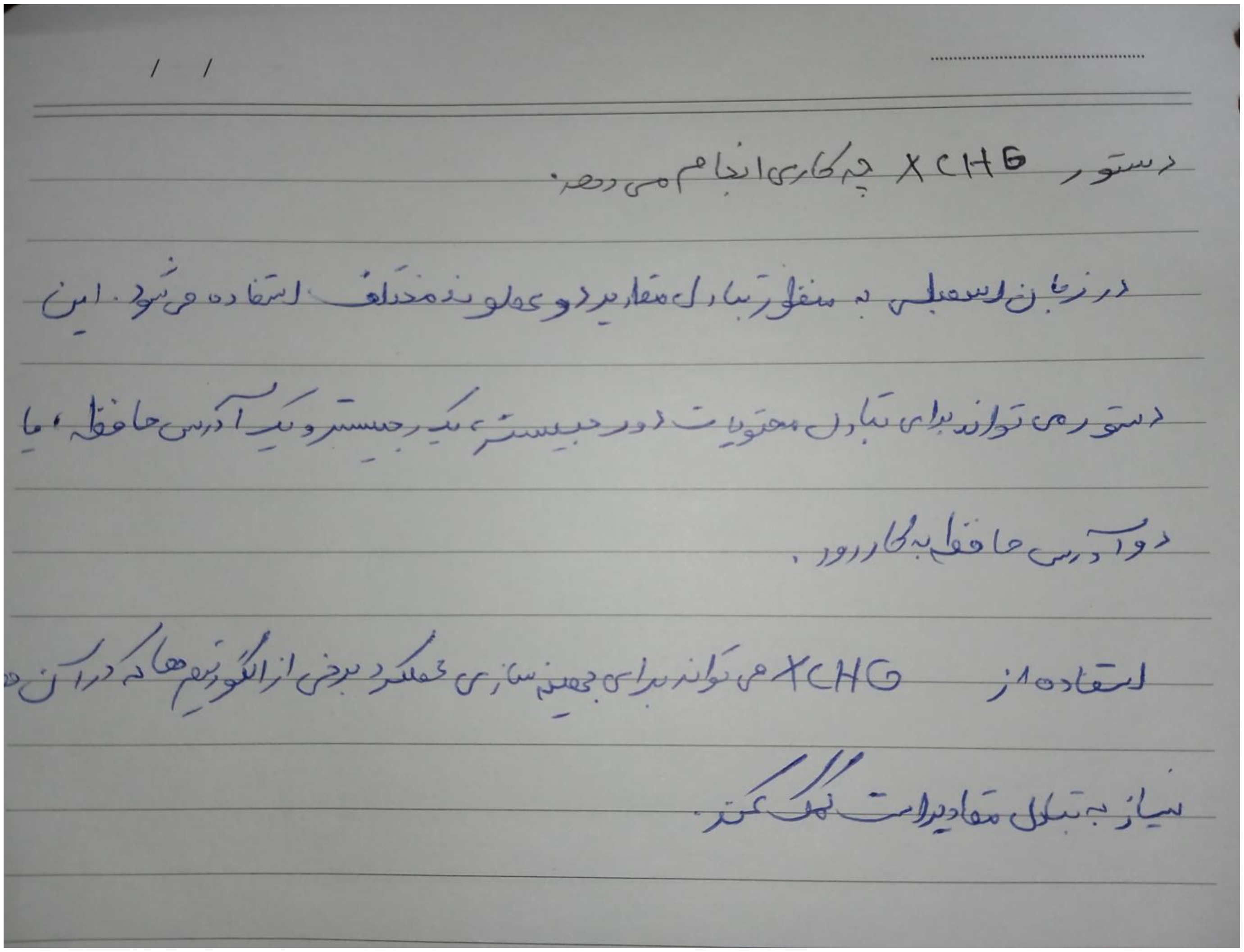


/ /

دستور LEA چه کاری انجام می دهد.

LEA در زبان اسمبلی و معماری X 86 وظیفه دریافت آدرس موثر یک متغیر را دارد و آن را یک ثبات خاص بارگذاری میکند. این دستور معمولاً برای محاسبه آدرس یک متغیر بدون بارگذاری مقدار خود آن متغیر استفاده می شود.

دستور LEA به شما اجازه می دهد تا به راحتی به آدرس های متغیرها و همچنین به آدرس محاسباتی دیگر دسترسی پیدا کنید. به عبارت دیگر این دستور آدرس یک متغیر یا یک مکان در حافظه را محاسبه می کند و نتیجه را در یک ثبات ذخیره می کند.



دستور XCHG چه کاری انجام می دهد؟

در زبان اسمبلی به منظور تبادل مقادیر دو عملوند مختلف استفاده می شود. این
دستور می تواند برای تبادل محتویات دو رجیستر، یک رجیستر و یک آدرس حافظه، یا
دو آدرس حافظه به کار رود.

استفاده از XCHG می تواند برای بهینه سازی عملکرد برخی از الگوریتم ها که در آن ها
نیاز به تبادل مقادیر است کمک کند.