

زبان ماشین و برنامه سازی سیستم

فصل دوازدهم: عملیات بر روی رشته اطلاعات

مدرس:

سید محمد بیدکی



دانشگاه خلیج فارس

دانشکده فنی و مهندسی جم

تابستان ۱۳۹۳

عناوین

- I. دستورات کار با رشته اطلاعات
- II. بیت پرچم DF
- III. تأثیر دستور REP
- IV. شرح دستورات کار با رشته اطلاعات
- V. دستور دستیابی به عناصر جدول (XLAT)

دستورات پردازش رشته اطلاعات

- **MOVS**: انتقال یک رشته اطلاعات به محل رشته دیگر
- **LODS**: بارکردن از یک رشته اطلاعات به ثبات **AL** یا **AX**
- **STOS**: ذخیره اطلاعات از **AL** یا **AX** به حافظه
- **CMPS**: مقایسه دو رشته اطلاعات
- **SCAS**: جستجوی اطلاعات ثبات **AL** یا **AX** در یک رشته اطلاعات
- **OUTS**: خروجی رشته اطلاعات از حافظه به پورت
- **INS**: ورود رشته اطلاعات از پورت به حافظه

دستورات پردازش رشته اطلاعات ...

- معمولاً دستورات رشته، عملیات را روی یک بایت یا یک کلمه تکرار می نمایند.
- آخر همه دستورات رشته میتواند کاراکترهای B یا W بیاید.
- B به معنای عملیات بایتی
- W به معنای عملیات کلمه ای
- آدرس رشته اطلاعات مبدأ در حافظه با آدرس ثباتهای DS:SI مشخص می شود.
- آدرس رشته اطلاعات مقصد، در حافظه با آدرس ثباتهای ES:DI مشخص می گردد.
- لذا در برنامه های به صورت EXE باید ثباتهای سگمنت داده اضافی ES و سگمنت داده DS به طور مناسب، مقداردهی اولیه شود.

تأثیر بیت پرچم جهت DF

- صفر یا یک بودن بیت پرچم DF در نحوه عملکرد دستورات رشته تأثیر دارد.
- $DF=0$ ، عملیات رشته از چپ به راست (اول به آخر): پس از اجرای دستورات رشته، آدرسهای موجود در ثباتهای SI و DI، به صورت صعودی (۱ یا ۲ واحد) اضافه می شود.
- $DF=1$ ، عملیات رشته از راست به چپ (آخر به اول): پس از اجرای دستورات رشته، آدرسهای موجود در ثباتهای SI و DI، به صورت نزولی (۱ یا ۲ واحد) کاهش می یابد.
- دستور **CLD**: صفر کردن بیت پرچم DF
- دستور **STD**: یک کردن بیت پرچم DF

تأثیر کلمه REP

- از این کلمه برای تکرار یک عملیات رشته استفاده می شود.
 - اگر قبل از دستورات رشته، کلمه REP نوشته شود، هر بار که دستور رشته اجرا می گردد، یک واحد از محتوای ثبات CX کسر شده، و تا موقعی که CX صفر نشود، عملیات رشته ادامه می یابد.
 - در صورتی که CX برابر صفر شد، عملیات رشته پایان می یابد.
- مثال:

```
MOV CX,10  
REP MOVSB
```

دستور انتقال رشته اطلاعات MOVSW

- این دستور به دو صورت MOVSB و MOVSW وجود دارد.
- این دستور باعث می شود که یک یا دو بایت از حافظه با آدرس موجود در SI، به محلی از حافظه با آدرس موجود در DI انتقال یابد و یک یا دو واحد به ثباتهای SI و DI اضافه (کسر) گردد.
- مثال: (کپی یک رشته در رشته دیگر)

```
MSG1 DB 'SALAM.....'
```

```
MSG2 DB 20 DUP(?)
```

```
...
```

```
CLD
```

```
LEA SI, MSG1
```

```
LEA DI, MSG2
```

```
MOV CX, 10
```

```
REP MOVSW
```

❖ **تذکر:** مقصد اطلاعات در سگمنت داده اضافی می باشد. لذا در برنامه های به صورت EXE حتماً ES باید مقداردهی شود.

دستور بارکردن از رشته اطلاعات LODS

```
ARRAY db 5,2,7,10,22,17  
MAX db ?
```

...

```
CLD  
LEA SI, ARRAY  
MOV MAX, 0  
MOV CX, 6  
L1:  
LODSB  
CMP AL, MAX  
JLE endLoop  
MOV MAX, AL  
endLoop:  
LOOP L1
```

- این دستور به دو صورت LODSB و LODSW وجود دارد.

- اجرای این دستور باعث می شود که یک یا دو بایت از رشته اطلاعات که آدرس آن در SI است، به ثبات AL یا AX منتقل شود و سپس یک یا دو واحد به ثبات SI اضافه (یا کم) شود.

- مثال: (اعداد یک آرایه را خوانده و بزرگترین آنها را بیابید)

دستور ذخیره کردن در یک رشته STOS

- این دستور به دو صورت STOSB و STOSW وجود دارد.
- اجرای این دستور باعث می شود که محتوای ثبات AL یا AX به یک یا دو بایت از رشته اطلاعات که آدرس آن در DI است، منتقل گردد و سپس یک یا دو واحد به ثبات DI اضافه (یا کم) شود.
- مثال: (محتوای ۵۰ کلمه حافظه از آدرس متغیر X به بعد را با صفر پر کنید)

```
X db 100 dup(?)
```

```
...
```

```
CLD
```

```
LEA DI, X
```

```
MOV CX, 50
```

```
MOV AX, 0
```

```
REP STOSW
```

تأثیر کلمات REPNE، REPE، REPZ و REPNE

- این کلمات در دستورات مقایسه دو رشته CMPS و جستجوی اطلاعات در یک رشته SCAS، به کار می روند و تقریباً مشابه REP عمل می کنند.
- کلمات REPZ و REPE معادلند و اگر قبل از دستورات رشته قرار گیرند، تا زمانی که دو آپرند دستور مساوی هستند ($ZF=1$) و CX مخالف صفر است، عملیات رشته ادامه می یابد.
- در صورتی که مقایسه صورت گیرد، به محض اینکه دو رشته غیرمساوی شوند و یا ثبات CX صفر شود، عملیات پایان می یابد.
- کلمات REPNE و REPZ نیز معادلند و عملکرد عکس فوق را دارند.

دستور مقایسه دو رشته اطلاعات CMPS

N1 db 'ALI REZA'

N2 db 'ALI REZA'

...

CLD

LEA SI, N1

LEA DI, N2

MOV CX, 8

REPE CMPSB

JNE less

MOV BH, 0

JMP exit

less:

JNB greater

MOV BH, 1

JMP exit

greater: MOV BH, 2

exit:

15

- این دستور به دو صورت CMPSB و CMPSW وجود دارد.

- اجرای این دستور باعث می شود یک یا دو بایت از اطلاعات که در خانه های حافظه با آدرس DS:SI و ES:DI هستند، مقایسه شوند و پس از هر بار اجرای این دستور، یک یا دو واحد به ثبات SI و DI اضافه (یا کم) شود.

- این دستور بر بیت های پرچم SF، AF، PF و CF تأثیر می گذارد.

- مثال: (نام دو نفر که در دو متغیر N1 و N2 تعریف شده اند را مقایسه کنید)

دستور جستجو در رشته اطلاعات SCAS

- این دستور به دو صورت SCASB و SCASW وجود دارد.

```
STR db 'HO...M?PNM'
```

```
...
```

```
CLD
```

```
LEA DI, STR
```

```
MOV AL, '?'
```

```
MOV CX, 20
```

```
REPNE SCASB
```

```
JNE notFound
```

```
MOV BH, 1
```

```
JMP exit
```

```
notFound:
```

```
MOV BH, 0
```

```
exit:
```

- اجرای این دستور باعث می شود بایت یا کلمه حافظه با آدرس ES:DI با محتوای ثبات AL یا AX مقایسه گردد. پس از هربار اجرای این دستور، یک یا دو واحد به ثبات DI اضافه (یا کم) شود.

- این دستور بر بیت‌های پرچم SF، OF، ZF، AF، PF و CF تأثیر می گذارد.

- مثال: (بررسی کنید در متغیر STR به طول ۲۰ بایت، کاراکتر ؟ وجود دارد یا خیر)

دستور ورودی رشته INS

- این دستور به دو صورت **INSB** و **INSW** وجود دارد.
- اجرای این دستور باعث می شود که یک یا دو بایت از پورتی که آدرس آن در ثبات **DX** است به حافظه با آدرس **ES:DI** منتقل گردد و سپس یک یا دو واحد به ثبات **DI** اضافه (یا کم) شود.
- مثال: (از دستگاه ورودی به آدرس ۴۵۰، تعداد ۱۰۰ کلمه را در آرایه **X** بخوانید)

```
X dw 100 DUP(?)
```

```
...
```

```
CLD
```

```
MOV DX, 450
```

```
LEA DI, X
```

```
MOV CX, 100
```

```
REP INSW
```

دستور خروجی رشته OUTS

- این دستور به دو صورت OUTSB و OUTSW وجود دارد.
- اجرای این دستور باعث می شود که یک یا دو بایت از اطلاعات حافظه با آدرس DS:SI به پورتی که آدرس آن در ثبات DX است، منتقل گردد و سپس یک یا دو واحد به ثبات SI اضافه (یا کم) شود.
- مثال: (۱۰۰ عنصر آرایه X را در دستگاه خروجی به آدرس ۴۵۰ ارسال کنید)

```
X dw 100 DUP(10)
```

```
...
```

```
CLD
```

```
MOV DX, 450
```

```
LEA SI, X
```

```
MOV CX, 100
```

```
REP OUTSW
```

دستور دستیابی به عناصر جدول XLAT

- شماره آفست یک عنصر از جدول در AL قرار دارد. این دستور باعث می شود که به آفست شروع جدول در حافظه که در BX قرار دارد مقدار AL اضافه شود ($BX + AL$) سپس محتوای خانه حافظه با آدرس $BX + AL$ به ثبات AL منتقل گردد.
- بیشتر برای طراحی جداول تبدیل کدها کاربرد دارد.
- مثال:

```
TABLE db 'A','B','C','D','E','F','G','H','I'
```

```
...
```

```
LEA BX, TABLE
```

```
MOV AL, 05
```

```
XLAT
```