عتید خدائی – ۴۰۰۰۷۹۶۳

نکات مهم

. برای اجرا پروژه، فقط کافی‌ست با داکر، دستور docker-compose up -d را بزنید، پس از مدتی پروژه روی پورت ۸۶۴۱ شما قابل استفاده است.

. هنگام تایپ کدهای میکروپروگرام، حتما دستور هر F را بصورت کاما از یکدیگر جدا کنید و میان آنان اسپیس نزنید. مثلا

FOLAN,BAHMAN,BISAR درست است و FOLAN, BAHMAN, BISAR غلط است.

. در دیگر قسمت‌های کد میکروپروگرام و کد اسمبلی نیازی به رعایت اسپیس نیست.

. نکته مهم دیگری به ذهنم نمی‌رسه :)

توضیحات مختصر

. ابتدا کدهای تایپ شده در داخل تکست‌باکس میکروپروگرم را گرفته، سپس لیبل و آدرس مدنظر را به یکدیگر مپ می‌کنیم تا در مرحله بعد آدرس دهی به راحتی بتوانیم آدرس‌هارا جایگذاری کنیم. پس از این مرحله وارد دیکود دستورات می‌شویم و خط به خط آنهارا با کمک دستور split جاوا اسکریپت می‌خوانیم و برای هر کلمه‌ای که پردازش می‌شود آدرس یا بیت مورد نظر آن را می‌نویسیم و در نهایت به ازای هر خط آنهارا در مموری مربوط به میکروپروگرم می‌نویسیم. در دو مرحله دیکد آدرس‌ها باید حواسمان به دستورات ORG باشد که می‌توانید باعث تغییر در آدرس خطوط بعدی شود.

. مانند قبل ورودی برنامه اسمبلی خود را گرفته و مرحله مپ کردن آدرس و لیبل‌ها به یکدیگر را انجام می‌دهیم. اینجا به هر خط که می‌رسیم با ۴ نوع دیتا ممکن است رو به رو شویم، یکی ORG یکی HLT یکی متغیر و دیگری دستور. اگر ORG بود، شماره آدرس مموری را برابر عدد رو به روی ORG می‌گذاریم، اگر HLT بود من در کدم FFFF وارد حافظه می‌کنم و اگر دستور بود به ازای هر کلمه‌ای که در آن است، آدرس آن لیبل را می‌آوریم، آدرس آن لیبل می‌تواند در حافظهٔ میکروپروگرام باشد (اگر دستور باشد) یا در همین حافظهٔ مموری (متغیر باشد) کلمهٔ سوم که نشان‌دهندهٔ INDIRECT هست را نیز می‌گیریم و دیکد می‌کنیم و در نهایت با همهٔ این دیکدها یک عدد ۱۶ بیتی ساخته‌ایم و آن را می‌توانیم به مموری در آدرس مورد نظر که از شماره خط دستورمان نسبت به ORG قبلی‌اش می‌آید.

. برای کامپایل ابتدا مقدار CAR را برابر با ابتدای دستور FETCH می‌گذارم و هربار آدرس درون CAR را می‌آرم و می‌رم به همون آدرس در داخل آدرس میکروپروگرم و دستور مورد نظر رو دیکد و اجرا می‌کنم. این شکلی تمام کدهای نوشته شده در برنامه اسمبلی اجرا می‌شوند.

(توضیحات بیشتر در ویدئو با مرور تمام توابع سورس کد آمده است)

کد میکروپروگرام استفاده شده در ویدئو:

```
ORG 0
ADD: NOP I CALL INDRCT
READ U JMP NEXT
ADD U JMP FETCH
ORG 4
LOAD: NOP I CALL INDRCT
READ U JMP NEXT
DRTAC U JMP FETCH
ORG 8
STORE: NOP I CALL INDRCT
ACTDR U JMP NEXT
WRITE U JMP FETCH
ORG 64
FETCH: PCTAR U JMP NEXT
READ,INCPC U JMP NEXT
DRTAR U MAP
INDRCT: READ U JMP NEXT
DRTAR U RET
END
```

کد اسمبلی استفاده شده در ویدئو:

```
ORG 0
LOAD X I
ADD Y
STORE RES
HLT
X, HEX 10
Y, HEX A
RES, HEX 0
ORG 10
HEX 10
END
```