COMP2300: 어셈블리어 프로그래밍

(강의록)

내용

[A 소개 3](#_Toc136321991)

[B 디지털 컴퓨터 구조(기초) 4](#_Toc136321992)

[C 기계어 프로그래밍 5](#_Toc136321993)

[D 6502 어셈블리 기초 문법 6](#_Toc136321994)

[E 산술 논리 연산 7](#_Toc136321995)

[F 조건문 8](#_Toc136321996)

[1. cmp 8](#_Toc136321997)

[1.1. 8](#_Toc136321998)

# 소개

# 디지털 컴퓨터 구조(기초)

# 기계어 프로그래밍

# 6502 어셈블리 기초 문법

# 산술 논리 연산

# 조건문

## 조건문

어셈블리어에는 엄밀한 의미의 if문은 없다. 고수준 언어의 if문은 어셈블리 수준의 여러 연산을 하나로 합친 고마운 기능이다. 고수준 언어의 if 문은 조건의 참/거짓에 따라 특정한 코드 블록을 실행하는 형태이다. 따라서 이를 어셈블리어로 구현하기 위해서는 특정 메모리 주소로 이동하는, 즉 실행 흐름을 바꾸는/분기하는 니모닉이 필요하다. 이 때 jmp를 쓴다.

아래는 어셈블리어로 작성한 일반적인 조건문이다.

고수준 언어의 if 문과 논리는 유사하지만, 코드로 구현하는 것은 조금 다르다. 고수준 언어에서는 if 문이 참인 경우 다음 블록을 실행한다라고 생각하고 작성하지만, 어셈블리어에서는 특정 조건이 거짓인 경우 특정 주소로 점프한다라고 생각하고 작성해야한다. 단, 반복해서 말하지만, 논리는 유사하다. jmp 2개가 필요, 조건문이 뒤집힌 느낌

참/거짓을 판단하는 명령어는?

텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 도표이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명

위 두 단계는 각각의 니모닉으로 표현됨

1 단계에 대응하는 니모닉은 cmp임. 2단계에 대응하는 니모닉은 상태 레지스터를 이용하고 몇 가지 있음

cmp는 뺄셈을 이용함.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 소프트웨어이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 웹사이트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 웹사이트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 웹사이트, 웹 페이지이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 웹사이트, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명

cmp는 부호 없는 경우에만 비교가 가능하다

텍스트, 스크린샷, 사람, 의류이(가) 표시된 사진

자동 생성된 설명

특별한 언급이 없다면 유효한 동작은 언제나 unsigned인 경우로 한정된다.

signed 연산에는 언제나 추가적인 고민이 필요하다.

텍스트, 스크린샷, 사람, 웹사이트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 의류, 웹사이트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 웹사이트이(가) 표시된 사진

자동 생성된 설명텍스트, 의류, 스크린샷, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 의류이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명

어셈블러가 bne $8003을 D0 FA로 어셈블함.

FA는 -6으로, 0x8007 주소로부터 -6 상대 주소인 0x8003으로 점프하라는 것으로 잘 어셈블된 것을 확인할 수 있음.

텍스트, 스크린샷, 사람, 의류이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 의류, 웹사이트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 사람, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 웹사이트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 웹사이트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 의류이(가) 표시된 사진

자동 생성된 설명

nums에 양수가 1개, 음수가 3개 이기 때문에, 점프 동작이 적은 bpl을 사용한 코드가 더 빠를 수 있다. 이처럼 데이터의 분포를 미리 알고 있을 경우, 그 데이터를 처리하기에 적합한 코드를 사용하여 속도를 증가시킬 수 있다.

텍스트, 스크린샷, 웹사이트, 사람이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 의류, 사람이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 사람, 의류이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명