



EDDI

Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 3기

2021. 12. 18

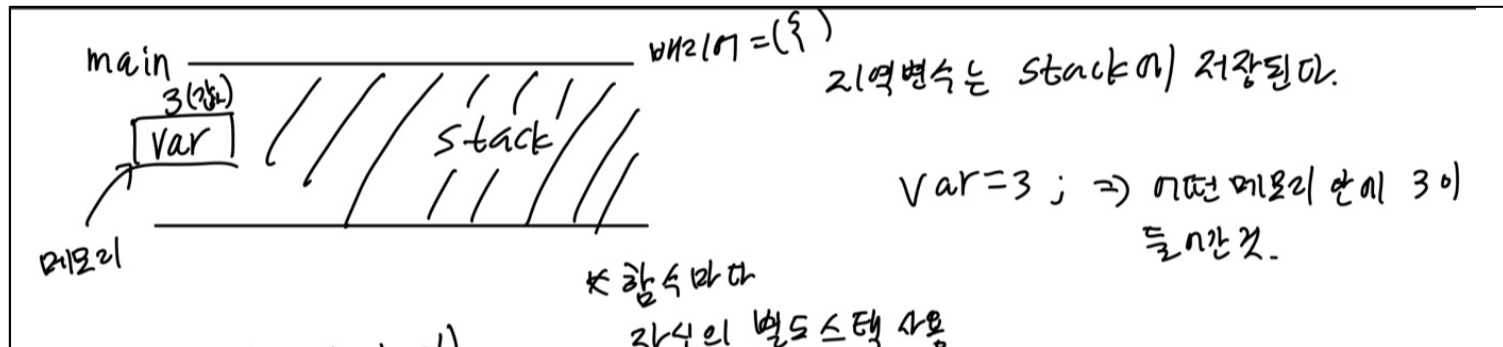
김원석

CONTENTS

- 변수
- 데이터 타입
- Vi 편집기
- GCC
- GDB
- 디버깅 흐름

변수란?

- 특정한 데이터 타입을 가지고 있고 정보를 저장할 수 있는 메모리 공간
- 함수마다 자신의 별도 스택을 사용하고, 지역변수는 Stack에 저장된다.
- 변수의 이름은 주소에 해당한다.
- PC에서 메모리라 부르는 것이 DRAM이고, DRAM에 데이터가 올라가는 것 자체가 변수에 해당한다.



1bit?

- 0혹은 1을 표현할 수 있는 컴퓨터에서 가장 작은 단위
- 1 byte = 8 bit

데이터 타입의 종류

- char = 1byte
- short = 2byte
- int = 4byte
- float = 4byte
- double = 8byte
- long long int = 12byte
- long double = 16 byte

- a혹은 i를 눌러 명령모드에서 입력모드로 전환할 수 있다.
- 명령모드에서 yy는 복사, p는 붙여넣기
- :set nu를 명령모드에서 입력해 줄 번호를 붙일 수 있다.
- 명령모드에서 dd는 한 줄을 지운다.
- y숫자y, d숫자d로 숫자에 해당하는 만큼 복사하거나 지울 수 있다.
- :%s/strawberry/melon/g 를 통해 모든 strawberry를 melon으로 바꿀 수 있다.
(:%s/[원본단어]/[바꿀 단어]/[옵션])
- ctrl + f는 page up, ctrl + b는 page down
- :줄번호 를 통해 특정 줄로 이동할 수 있다.

실행하기

- gcc [파일명].c 를 입력하면 해당 C파일이 컴파일 된다.
- ./[실행파일명].out를 입력하면 실행할 수 있다.

실행파일명 지정하기

- gcc -o [만들파일명] [소스파일명].c

※참고

- .는 현재위치를 의미하고 ..는 상위 디렉토리를 의미한다.

GDB 사용하기

- -g 옵션을 추가해서 컴파일 한다.
- 컴파일한 파일을 gdb [실행파일명] 명령어로 실행한다.

GDB 기능

- p [변수명] : 해당 변수 값 출력([\$레지스터]는 해당 레지스터 값 출력)
- x/[옵션] [메모리주소(또는 레지스터)] : 메모리 검사할 때 사용
- s : C언어 단위로 이동한다.(함수 호출 시 함수 내부로 이동)
- n : C언어 단위로 이동한다.(함수 호출 시 건너 뛴)

- si : 기계어 단위로 이동(함수 호출 시 함수 내부로 이동)
- ni : 기계어 단위로 이동(함수 호출 시 건너 뛴)
- b [함수명] : 특정 함수에 breakpoint 지정
- shell clear : 화면 지우기
- r(run) : 프로그램 구동
- l : C코드 단위로 코드 보기
- disas [함수명(주소)] : 해당 함수를 기계어 코드로 디버깅한다.
- info registers : 실제 CPU에 해당하는 연산용 레지스터가 나온다.

※ 화살표에 해당하는 라인은 아직 실행하지 않은 상태이다.

1. 디버깅 옵션을 주고 C소스파일을 컴파일
2. GDB를 이용해 실행파일 디버깅 시작
3. breakpoint를 분석하고자 하는 지점에 배치
4. r을 통해 프로그램을 시작하면 breakpoint위치에서 정지한다.
5. n이나 s를 활용해 실행
6. 필요하다면 disas를 이용해 어셈블리 코드를 보면서 실행
7. p를 통해 원하는 값을 살펴보고 x를 통해 메모리를 살펴보며 분석
8. 종료하고자 할 때 q를 눌러 분석 종료