



EDDI

Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 4기

2022. 09. 15

진동민

학습목표 & 1회차 날짜

학습목표

- main 함수의 반환값 이해하기
- 리눅스를 다루기 위한 쉘 기초 명령어 배우기
- 프로그램을 작성하기 위한 vi 편집기 사용법 배우기

수업 날짜

2022-08-20 (토) 오후 6시~9시

목차

- 1) C 언어 기초
- 2) 리눅스 쉘 기본 명령어
- 3) vi 편집기 사용법
- 4) 노선에 이슈 기록 방법

1) main 함수의 반환형을 무엇으로 해야하는가

void main(void) 를 적는 경우가 있는데 펌웨어 코드와 다르게 **애플리케이션 프로그램을 작성하는 경우에는 반환형을 void 로 작성하면 안된다.**

그 **이유**는 무엇일까?

- main 함수의 리턴값이 운영체제로 가기 때문이다. 실제 이 프로그램을 실행해서 관리하는 부모 프로세스가 해당 프로세스가 정상 종료되었는지 비정상 종료되었는지 파악하기 위한 용도 리턴값을 사용한다.

(참고 - 출처: C++ 기초 플러스 6판, p.59)

- 많은 운영 체제가 프로그램이 리턴하는 값을 처리할 수 있다.
- 예를 들어, 프로그램을 실행하고, 프로그램이 리턴하는 종료값(exit value)을 검사하는 Unix의 셸 스크립트(shell script)나 DOS의 배치(batch) 파일을 작성할 수 있다.
- 일반적으로 종료값이 0이면 프로그램이 성공적으로 실행되었다는 것을 뜻하고, 종료값이 0이 아닌 다른 값이면 문제가 발생했다는 것을 뜻한다.
- 그러므로 가령 파일을 열다 실패했을 경우에 0이 아닌 값을 리턴하는 C++ 프로그램을 작성할 수 있다.
- 그리고 나서 그 프로그램을 실행시키고, 만약 실행이 실패했을 경우에는 다른 작업을 처리하는 셸 스크립트나 배치 파일을 설계할 수 있다.

***결론 :** 운영체제의 경우 프로그램의 리턴값을 사용할 수 있으므로 애플리케이션을 개발할 때는 main 함수의 반환형을 void로 작성하면 안된다.

2) 변수를 선언하는 방법

1. 데이터 타입을 적는다.
2. 변수 이름을 적는다.
3. 필요하다면 초기화를 한다.
 - 초기화 방법으로는 ' = 넣고 싶은 숫자' 형태로 작성한다.

3) 변수란 무엇인가

특정한 데이터 타입을 저장할 수 있는 메모리(DRAM, 상황에 따라 SRAM이 될 수 있음) **공간**

```
int num = 3;
```

carbon
carbon.now.sh

<https://carbon.now.sh/>
사용

위 코드는 실제로 어셈블리어로 mov 명령어로 바뀌게 되므로 num은 3이다라는 개념이 아니라 숫자 3을 num에 해당하는 메모리 공간에 배치(저장)한다라는 개념으로 연결된다.

4) main 함수의 리턴값을 직관적으로 작성하자

- 앞서 1)의 참고에서 보았듯이 리턴값으로 프로그램의 종료 상태를 확인할 수 있다.

```
1 #include <stdio.h>
2
3 #define ERROR      -1
4 #define SUCCESS    0
5 #define MEMORY_ERROR 2
6
7 int main(void)
8 {
9     // 생략
10
11     return SUCCESS;
12 }
```

carbon
carbon.now.sh

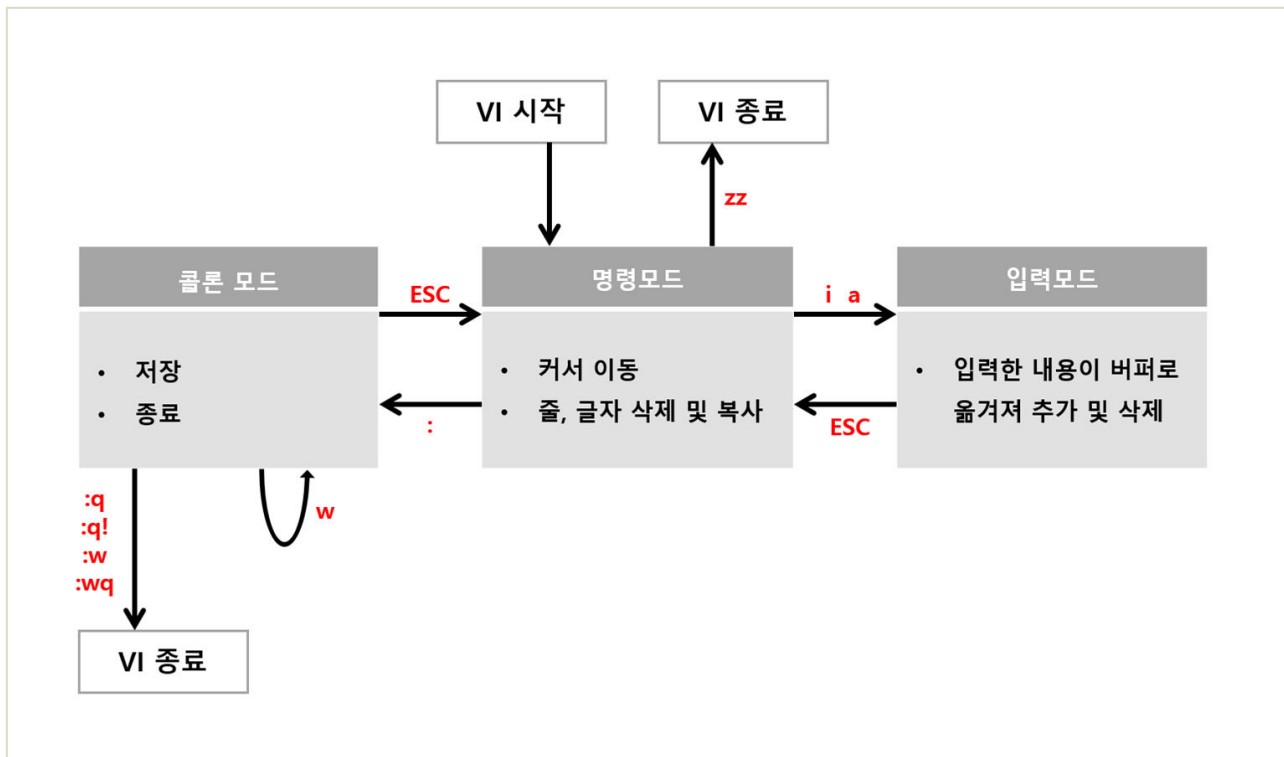
- 만약 위의 코드와 상반되게 반환값으로 숫자를 적으면 해당 숫자가 무엇을 하는지 어떤 의미를 갖는지 파악하기 위해 상당히 많은 비용을 소모하게 될 수 있다.
- 그러므로 **맨 위의 #define 혹은 enum 등을 활용하여 사람이 알아볼 수 있게 만들어주는 것이 중요하다.**

- mkdir(make directory): 디렉토리를 생성
 - 예) mkdir eddi ← 'eddi' 라는 이름의 디렉토리 생성
- ls(list): 현재 위치에 존재하는 파일과 디렉토리를 목록을 출력
 - 예) ls -al ← 숨겨진 파일을 포함하여 파일들을 나열할 때 자세히 출력
- pwd(print working directory): 현재 디렉토리 위치를 출력
- cd(change directory): 디렉토리 변경(이동)
 - 예) cd / ← 현재 위치에서 최상위(root 디렉토리)로 이동
- nautilus: GUI 상태로 디렉토리를 볼 수 있음
 - 예) nautilus ← 홈 디렉토리를 기준으로 파일 관리자 창을 열음
 - 예) nautilus . ← 현재 디렉토리 위치를 기준으로 파일 관리자 창을 열음

(참고)

- . ← 현재 디렉토리
- .. ← 상위 디렉토리
- / ← root 디렉토리(최상위)

vi 편집기 사용법



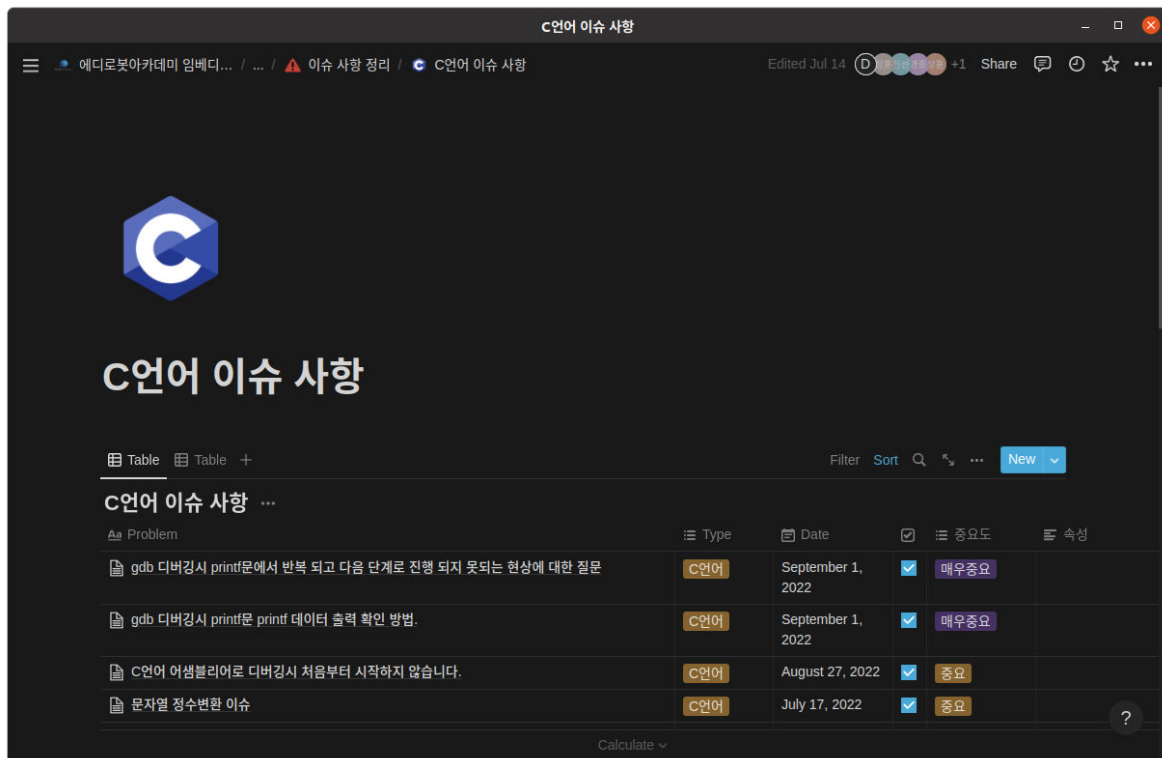
출처: <https://velog.io/@zeesoo/Linux-vi-편집기-사용법-및-명령어>

커맨드 모드에서 중요하다고 생각되는 명령어

- `':wq'` ← 저장하고 나가기
- `y숫자y` ← (커서가 위치한 라인을 포함하여) 숫자만큼 라인을 복사
- `d숫자d` ← (커서가 위치한 라인을 포함하여) 숫자만큼 라인을 삭제
- `u` ← 실행 취소
- `ctrl + R` ← 다시 실행
- `p` ← (커서 바로 아래 줄에) 붙여넣기

노선에 이슈 기록 방법

1) 작성할 주제에 알맞는(또는 가까운) 페이지로 들어온다.



C언어 이슈 사항

Table | Table +

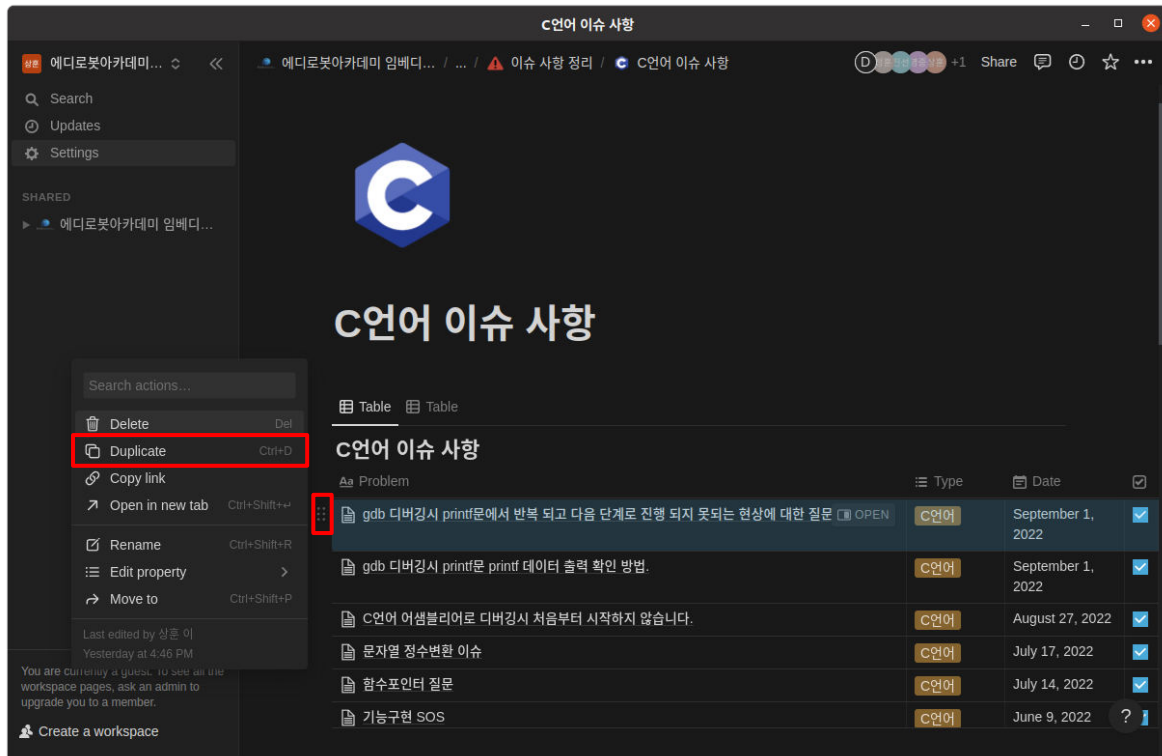
Filter Sort Q ... New

Problem	Type	Date	✓	중요도	속성
gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문	C언어	September 1, 2022	✓	매우중요	
gdb 디버깅시 printf문 printf 데이터 출력 확인 방법.	C언어	September 1, 2022	✓	매우중요	
C언어 어셈블리어로 디버깅시 처음부터 시작하지 않습니다.	C언어	August 27, 2022	✓	중요	
문자열 정수변환 이슈	C언어	July 17, 2022	✓	중요	

Calculate

노션에 이슈 기록 방법

2) 페이지내에서 최상단에 있는 페이지를 복제한다.

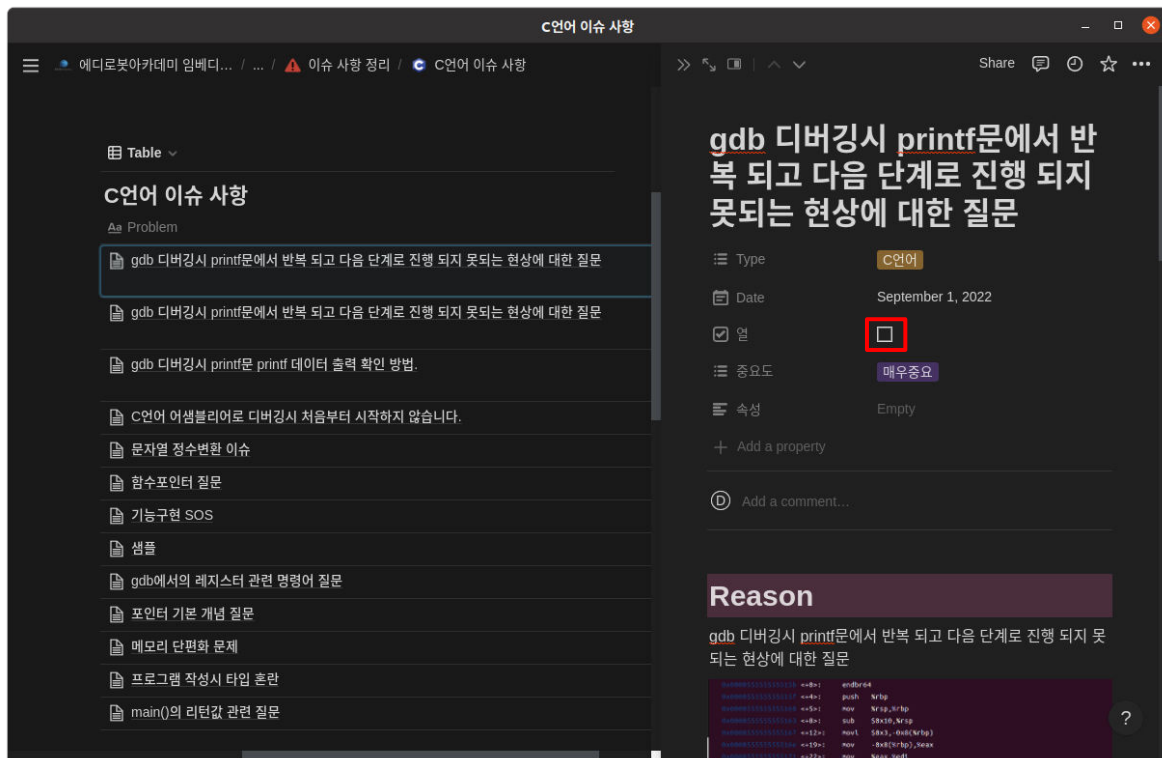


The screenshot shows a Notion workspace titled "C언어 이슈 사항". The left sidebar contains a search bar and navigation links. The main content area displays a table of issues. A context menu is open over the first issue, with the "Duplicate" option highlighted in red. The table has columns for "Problem", "Type", "Date", and a checkbox.

Problem	Type	Date	
gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문	C언어	September 1, 2022	<input checked="" type="checkbox"/>
gdb 디버깅시 printf문 printf 데이터 출력 확인 방법.	C언어	September 1, 2022	<input checked="" type="checkbox"/>
C언어 어셈블리어로 디버깅시 처음부터 시작하지 않습니다.	C언어	August 27, 2022	<input checked="" type="checkbox"/>
문자열 정수변환 이슈	C언어	July 17, 2022	<input checked="" type="checkbox"/>
함수포인터 질문	C언어	July 14, 2022	<input checked="" type="checkbox"/>
기능구현 SOS	C언어	June 9, 2022	<input type="checkbox"/>

노선에 이슈 기록 방법

3) 복제된 페이지를 클릭하여 체크를 해제하고, 내용을 지우고 기록한다.



The screenshot displays the EDDI issue tracking system interface. On the left, a sidebar shows a list of issues under the heading "C언어 이슈 사항". The selected issue is "gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문". The right pane shows the details of this issue, including its title, type (C언어), date (September 1, 2022), and a red square icon. The "Reason" section contains the text "gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문" and a code snippet.

C언어 이슈 사항

Table

C언어 이슈 사항

Problem

- gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문
- gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문
- gdb 디버깅시 printf문에서 printf 데이터 출력 확인 방법.
- C언어 어셈블리어로 디버깅시 처음부터 시작하지 않습니다.
- 문자열 정수변환 이슈
- 함수포인터 질문
- 기능구현 SOS
- 샘플
- gdb에서의 레지스터 관련 명령어 질문
- 포인터 기본 개념 질문
- 메모리 단편화 문제
- 프로그램 작성시 타입 혼란
- main()의 리턴값 관련 질문

gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문

Type: C언어

Date: September 1, 2022

열: ☒

중요도: 매우중요

속성: Empty

+ Add a property

① Add a comment...

Reason

gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문

```
0000011111111111: <<0>: endbr04
0000011111111111: <<4>: push %r10
0000011111111111: <<5>: mov %rsp,%bp
0000011111111111: <<6>: sub $0x10,%rsp
0000011111111111: <<7>: movl $0x3,0x0(%rsp)
0000011111111111: <<8>: mov -0x8(%bp),%eax
0000011111111111: <<9>: jmp %eax
```

노선에 이슈 기록 방법



3) 강사님의 답변이 달리고 해결되었으면 체크한다.

c언어 이슈 사항

☰
🔍 에디로봇아카데미 일베디... / ... / ⚠️ 이슈 사항 정리 / 🔄 C언어 이슈 사항

Table ▾

c언어 이슈 사항

Problem

gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문

gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문

gdb 디버깅시 printf문 printf 데이터 출력 확인 방법.

C언어 어셈블리어로 디버깅시 처음부터 시작하지 않습니다.

문자열 정수변환 이슈

함수포인터 질문

기능구현 SOS

샘플

gdb에서의 레지스터 관련 명령어 질문

포인터 기본 개념 질문

메모리 단편화 문제

프로그램 작성시 타입 혼란

main()의 리턴값 관련 질문

Type Date September 1, 2022
 열 📧
 중요도 매우중요
 속성 Empty
 + Add a property
 @ Add a comment...

Reason

gdb 디버깅시 printf문에서 반복 되고 다음 단계로 진행 되지 못하는 현상에 대한 질문

```

0000000000000000 <&b>=0: 0x0:
0000000000000000 0x40: push %rbp
0000000000000000 0x50: mov %rsp,%rbp
0000000000000000 0x60: sub $0x18,%rsp
0000000000000000 0x70: movl $0x3,-0x4(%rip)
0000000000000000 0x80: -0x4(%rip),%eax
mov %eax,%eax
                    
```