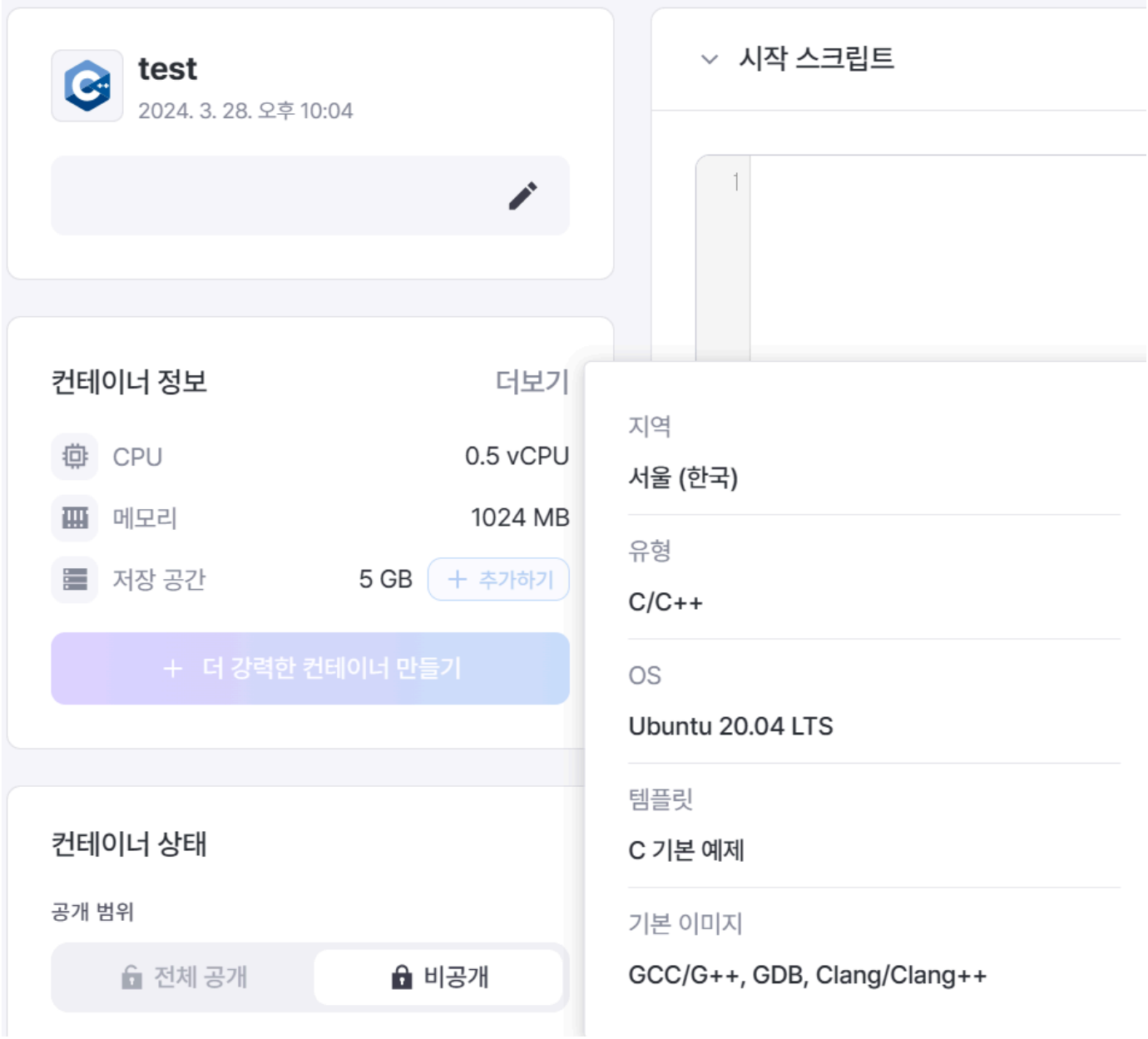
WHL 컴퓨터구조I

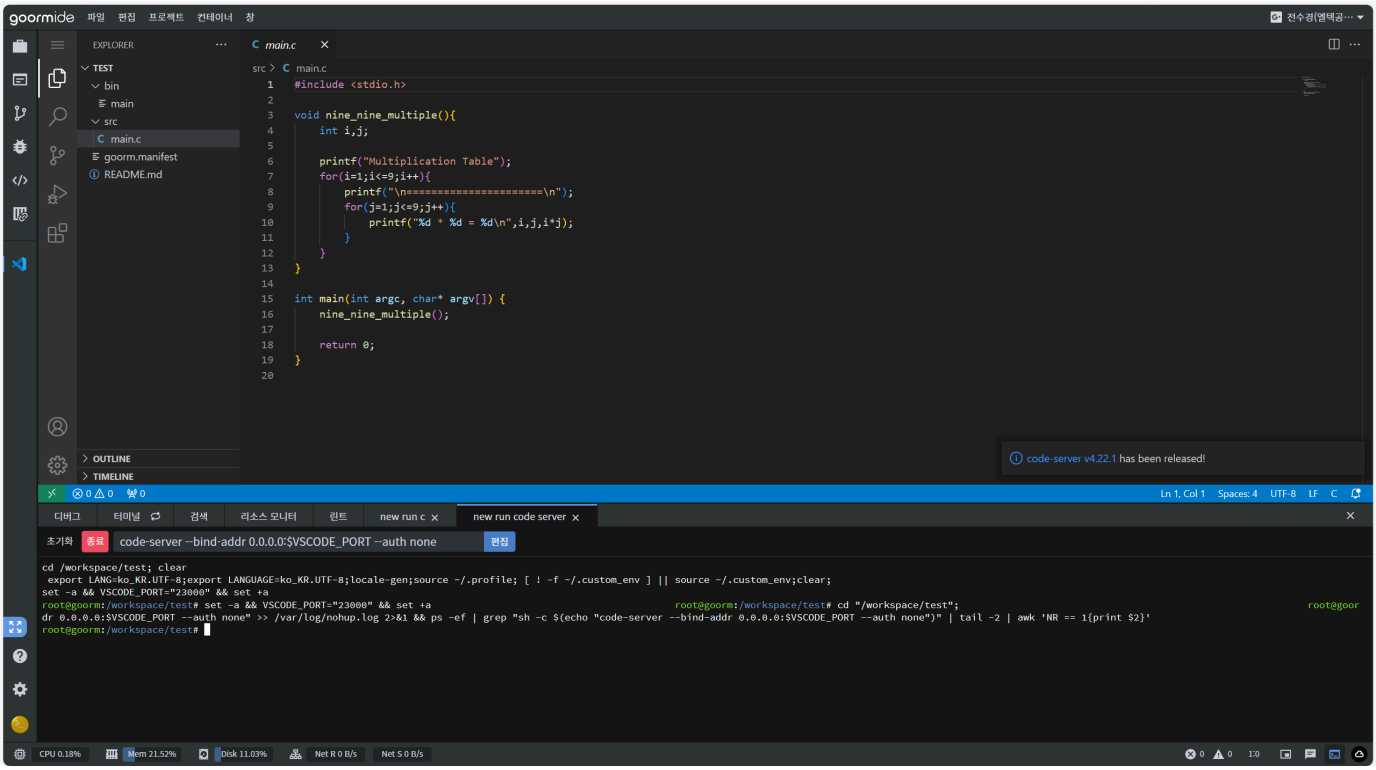
전수경 · 약 13시간 전

WHS

목록 보기 1/1

1. 구름IDE로 리눅스 개발환경 만들기


test
2024. 3. 28. 오후 10:04
시작 스크립트
1
컨테이너 정보
더보기
CPU
0.5 vCPU
메모리
1024 MB
저장 공간
5 GB
추가하기
더 강력한 컨테이너 만들기
지역
서울 (한국)
유형
C/C++
OS
Ubuntu 20.04 LTS
템플릿
C 기본 예제
기본 이미지
GCC/G++, GDB, Clang/Clang++
컨테이너 상태
공개 범위
전체 공개
비공개



2. 오버플로 예제를 언더 플로로 바꿔서 해보기 -CHAR_MIN의 값에서 -1

(1) 코드

```c
#include <stdio.h>
#include <limits.h> #<limits.h>는 정수형 자료형의 유효 범위를 확인할 수 있는 헤더

int main(){
    char value= CHAR_MIN;
    printf("Original value: %d\n", value);
    value=value - 1;
    printf("Value after adding 1: %d\n", value);

    return 0;
}
```

- CHAR_MIN은 <limits.h>에 정의된 CHAR의 최솟값으로 -128이다.(char 자료형은 -128~127)
- 이때 value 값을 한 번 더 빼면서 -129가 되는 것이 아니라, char의 최솟값보다 작으므로 언더플로가 일어나고 127값이 나올 것이다.

(2) 실행 결과

방법1.

```
root@goorm:/workspace/test/WHS_1# gcc -o main main.c -g
root@goorm:/workspace/test/WHS_1# gdb main
GNU gdb (Ubuntu 10.2-0ubuntu1~20.04~1) 10.2
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html
>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
--Type <RET> for more, q to quit, c to continue without paging--
Type "apropos word" to search for commands related to "word"...
Reading symbols from main...
(gdb) b main
Breakpoint 1 at 0x1155: file main.c, line 5.
```

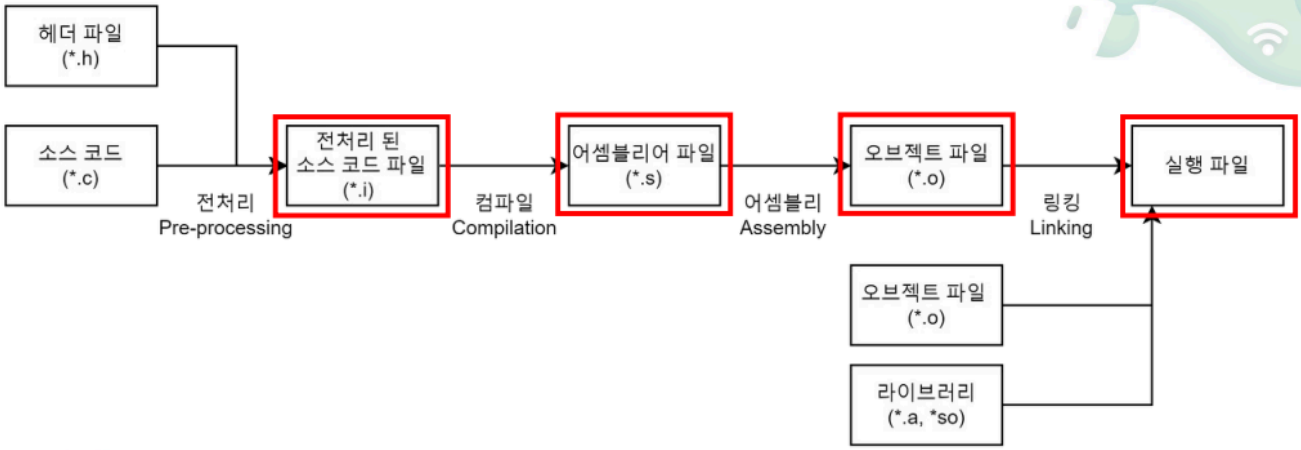
```
(gdb) b main
Breakpoint 1 at 0x1155: file main.c, line 5.
(gdb) r
Starting program: /workspace/test/WHS_1/main

Breakpoint 1, main () at main.c:5
5           char value= CHAR_MIN;
(gdb) n
6           printf("Original value: %d\n", value);
(gdb) n
Original value: -128
7           value=value - 1;
(gdb) p/t value
$1 = 10000000
(gdb) n
8           printf("Value after adding 1: %d\n", value);
(gdb) p/t value
$2 = 1111111
(gdb) p value
$3 = 127 '\177'
(gdb)
```

- $1=10000000은 이진수로 나타내서 -128, $2=1111111(01111111)는 127이다. 오버플로가 일어나서 -1을 해도 -129가 아닌 127로 나타난다.

방법2

```
root@goorm:/workspace/test/WHS_1# ./main
Original value: -128
Value after adding 1: 127
```

3. C언어가 기계어가 되는 과정



WHS 컴퓨터구조I 수업 pdf 중

(1) 전처리 된 소스 코드 파일 생성 및 확인

```
root@goorm:/workspace/test/WHS_1# gcc -E test1.c -o test1.i
```

```
34  # 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 1 3 4
35  # 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4
36
37  # 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4
38  typedef long unsigned int size_t;
39  # 34 "/usr/include/stdio.h" 2 3 4
40
41
42  # 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 1 3 4
43  # 40 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 3 4
44  typedef __builtin_va_list __gnuc_va_list;
45  # 37 "/usr/include/stdio.h" 2 3 4
46
47  # 1 "/usr/include/x86_64-linux-gnu/bits/types.h" 1 3 4
48  # 27 "/usr/include/x86_64-linux-gnu/bits/types.h" 3 4
49  # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
50  # 28 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
51  # 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
52  # 29 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
53
54
55  typedef unsigned char __u_char;
56  typedef unsigned short int __u_short;
57  typedef unsigned int __u_int;
58  typedef unsigned long int __u_long;
59
60
61  typedef signed char __int8_t;
62  typedef unsigned char __uint8_t;
63  typedef signed short int __int16_t;
64  typedef unsigned short int __uint16_t;
65  typedef signed int __int32_t;
66  typedef unsigned int __uint32_t;
```

```
34  # 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 1 3 4
35  # 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4
36
37  # 209 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stddef.h" 3 4
38  typedef long unsigned int size_t;
39  # 34 "/usr/include/stdio.h" 2 3 4
40
41
42  # 1 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 1 3 4
43  # 40 "/usr/lib/gcc/x86_64-linux-gnu/9/include/stdarg.h" 3 4
44  typedef __builtin_va_list __gnuc_va_list;
45  # 37 "/usr/include/stdio.h" 2 3 4
46
47  # 1 "/usr/include/x86_64-linux-gnu/bits/types.h" 1 3 4
48  # 27 "/usr/include/x86_64-linux-gnu/bits/types.h" 3 4
49  # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
50  # 28 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
51  # 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
52  # 29 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
53
54
55  typedef unsigned char __u_char;
56  typedef unsigned short int __u_short;
57  typedef unsigned int __u_int;
58  typedef unsigned long int __u_long;
59
60
61  typedef signed char __int8_t;
62  typedef unsigned char __uint8_t;
63  typedef signed short int __int16_t;
64  typedef unsigned short int __uint16_t;
65  typedef signed int __int32_t;
66  typedef unsigned int __uint32_t;
```

- test1.i 파일 일부

(2) 어셈블리어 파일 생성,오브젝트 파일 생성

```
root@goorm:/workspace/test/WHS_1# gcc -E test1.c -o test1.i
root@goorm:/workspace/test/WHS_1# gcc -S main.i -o main.s
```

```
root@goorm:/workspace/test/WHS_1# gcc -S test1.i -o test1.s
root@goorm:/workspace/test/WHS_1# gcc -c test1.s -o test1.o
```

```
    .file   "test1.c"
    .text
    .section    .rodata
.LC0:
    .string "Hello World!"
    .text
    .globl  main
    .type   main, @function
main:
.LFB0:
    .cfi_startproc
    endbr64
    pushq   %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    leaq    .LC0(%rip), %rdi
    call    puts@PLT
    movl    $0, %eax
    popq    %rbp
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE0:
    .size   main, .-main
    .ident  "GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0"
    .section    .note.GNU-stack,"",@progbits
    .section    .note.gnu.property,"a"
    .align 8
    .long    1f - 0f
    .long    4f - 1f
    .long    5
0:
    .string  "GNU"
1:
    .align 8
    .long    0xc0000002
    .long    3f - 2f
2:
    .long    0x3
3:
    .align 8
4:
```

- test1.s 파일

- test1.o 파일-> 기계어이기 때문에 파일 내용 확인이 안 됨

```
root@goorm:/workspace/test/WHS_1# file *
main:    ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically
 linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=26a5d226d6e5
16291e290d0a5586a9ebb4d1869a, for GNU/Linux 3.2.0, with debug_info, not stri
pped
main.c:  C source, ASCII text
test1.c: C source, ASCII text
test1.i: C source, ASCII text
test1.o: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped
test1.s: assembler source, ASCII text
```

- test1.o는 ELF 파일로 유닉스, 유닉스 시반 시스템에서 사용되는 공통 바이너리 파일 포맷
- LSB: Little Endian 바이트 순서
- relocatable: 재배치 가능=링커에 의해 다른 오브젝트 파일이나 공유 lib와 연결 가능
- 바이너리 형태

(3) obdjump 사용해 오브젝트 파일 확인하기

```
root@goorm:/workspace/test/WHS_1# objdump -d test1.o

test1.o:     file format elf64-x86-64


Disassembly of section .text:

0000000000000000 <main>:
   0:   f3 0f 1e fa             endbr64
   4:   55                      push   %rbp
   5:   48 89 e5                mov    %rsp,%rbp
   8:   48 8d 3d 00 00 00 00    lea    0x0(%rip),%rdi        # f <main+0xf>
   f:   e8 00 00 00 00          callq  14 <main+0x14>
  14:   b8 00 00 00 00          mov    $0x0,%eax
  19:   5d                      pop    %rbp
  1a:   c3                      retq
```

(4) Linking

```
root@goorm:/workspace/test/WHS_1# gcc test1.o -o test1
```

- test1 파일 생성

```
root@goorm:/workspace/test/WHS_1# ./test1
Hello World!
```

- test1 실행 및 결과

```
root@goorm:/workspace/test/WHS_1# file *
main:    ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=26a5d226d6e516291e290d0a5586a9ebb4d1869a, for GNU/Linux 3.2.0, with debug_info, not stripped
main.c:  C source, ASCII text
test1:   ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=dc35fc3d1674f1a20f64a3652aadbbc6054d250f, for GNU/Linux 3.2.0, not stripped
test1.c: C source, ASCII text
test1.i: C source, ASCII text
test1.o: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped
test1.s: assembler source, ASCII text
```

- 오브젝트 파일과 실행파일 차이

전수경

보안과 개발 그 사이..

0개의 댓글

댓글을 작성하세요

댓글 작성