컴파일러 수업

- 디버깅하기

경민기, 2025-12-09

디버깅 하는 이유 (1)

컴파일러 내부는 외부에서 안 보임

컴파일러는 “소스 → AST → IR → Assembly”로 가지만 겉으로는 오직 결과물(assembly)만 보임 중간에서 무슨 일이 일어나는지 알 수 없으면 버그의 위치를 추적할 수 없다.

파이썬 언어의 python 구현물인 pypy 프로젝트의 경우, 디버깅 작업에 7년 걸림

https://namu.wiki/w/PyPy

디버깅하는 이유 (2)

재귀 호출/스택 프레임/스코프 오류는 눈으로 추적하기 어렵다

factorial 같은 코드

```
return n * fact(n-1);
```

파싱은 잘 되었는데…

- 매 호출마다 **n의 오프셋**
- push/pop이 정상적으로 맞는지
- rdi ← 인자 전달이 올바른지
- rax ← return 값이 맞는지

이건 사람이 코드만 봐서는 추적이 어려움.

디버깅하는 이유 (3)

```
mov -8(%rbp), %rax
```

AST 노드가 잘못됐나? 연산자가 잘못됐나? 변수 오프셋이 잘못됐나?

=> 단번에 알 수 없음.

```
[DEBUG] load variable a (offset -16)
```

offset 이상 → 스택 할당 문제
load 이상 → AST 문제
연산 이상 → codegen 문제

바로 원인을 찾을 수 있음.

MiniC → x86-64 Assembly Compiler

- 입력: MiniC 소스 (`.minic`)
- 출력: x86-64 SysV ABI 어셈블리 (`.s`)
- 기능:
  a. 함수 정의 / 호출
  b. 지역 변수 스택 오프셋 관리
  c. 이항 연산 (+, -, *, /, < 등)
  d. 재귀 호출

- **디버그 로그 출력 기능(새로 추가)**

주요 특징 (New Features)

- 함수 생성 시작/끝 로그
- 변수 스택 오프셋 정보
- 함수 호출 정보
- 변수 로드/스토어 로그
- AST → Assembly 과정 상세 출력

실행

make clean

make

./minic_x86 test/add2.minic

make prog

./prog

echo $?

```
mingi-kyung@mingi-kyung-ThinkPad-E15-Gen-2:~/workspaces/compiler_class/13wk/minic_x86_debug$ ./minic_x86 test/add2.minic
[DEBUG] Opening input file: test/add2.minic
[DEBUG] Redirecting output to out.s
[DEBUG] Starting parsing
[DEBUG] Starting code generation
[DEBUG] Generating code for function: add
[DEBUG] Allocating locals for function: add
[DEBUG]   Param[0]: a at offset -8
[DEBUG]   Param[1]: b at offset -16
[DEBUG]   Local var: s at offset -24
[DEBUG] Total variables: 3, Stack size: 24
[DEBUG] Loading variable 'b' from offset -16 to rax
[DEBUG] Loading variable 'a' from offset -8 to rax
[DEBUG] Loading variable 's' from offset -24 to rax
[DEBUG] Finished generating code for function: add
[DEBUG] Generating code for function: main
[DEBUG] Allocating locals for function: main
[DEBUG]   Local var: x at offset -8
[DEBUG]   Local var: y at offset -16
[DEBUG]   Local var: result at offset -24
[DEBUG] Total variables: 3, Stack size: 24
[DEBUG] Generating function call to 'add'
[DEBUG] Loading variable 'x' from offset -8 to rax
[DEBUG] Loading variable 'y' from offset -16 to rax
[DEBUG] Function 'add' has 2 arguments
[DEBUG] Loading variable 'result' from offset -24 to rax
[DEBUG] Finished generating code for function: main
[DEBUG] Compilation completed successfully
```

실행결과

```
mingi-kyung@mingi-kyung-ThinkPad-E15-Gen-2:~/workspaces/compiler_class/13wk/minic_x86_debug$ ls
Makefile  README.md  build  include  minic_x86  out.s  parser  src  test
mingi-kyung@mingi-kyung-ThinkPad-E15-Gen-2:~/workspaces/compiler_class/13wk/minic_x86_debug$ make prog
gcc -no-pie out.s -o prog
/usr/bin/ld: warning: /tmp/cc2QHujM.o: missing .note.GNU-stack section implies executable stack
/usr/bin/ld: NOTE: This behaviour is deprecated and will be removed in a future version of the linker
mingi-kyung@mingi-kyung-ThinkPad-E15-Gen-2:~/workspaces/compiler_class/13wk/minic_x86_debug$ ls
Makefile  README.md  build  include  minic_x86  out.s  parser  prog  src  test
mingi-kyung@mingi-kyung-ThinkPad-E15-Gen-2:~/workspaces/compiler_class/13wk/minic_x86_debug$ ./prog
mingi-kyung@mingi-kyung-ThinkPad-E15-Gen-2:~/workspaces/compiler_class/13wk/minic_x86_debug$ echo $?
7
```