REPORT

[Hw 3]



과 목: 시스템소프트웨어 03

담당교수: 석문기 교수님

학 과: 컴퓨터공학과

학 번:2021111971

이 름:이재혁



문제 1

어셈블러의 역할, 오브젝트 파일, 정적 라이브러리가 무엇인지 짧게 설명 하시오.

1. 어셈블러

컴파일러에 의해 생성된 어셈블리 언어 코드를 프로세서가 실행할 수 있는 또는 바이너리 코드로 변환하는 역할을 합니다. C 프로그래밍에서 C 소스 코드가 어셈블리 코드로 컴파일 된후 어셈블러는 이 코드를 처리하여 해당 오브젝트 파일을 생성합니다.

2. 오브젝트 파일

C언어에서 오브젝트파일은 어셈블리 언어코드를 어셈블러가 바이너리 코드로 처리한 파일입니다. 기계어 코드가 포함되어 있지만 아직 완전한 실행 파일은 아닙니다. 실행 파일로 결합하는데 사용하는 기호(변수, 함수) 및 재배치 정보와 같은 정보가 포함됩니다.

3. 정적 라이브러리

정적 라이브러리는 오브젝트 파일의 집합입니다. C에서 함수 같은 재사용 가능한 코드가 포함된 정적 라이브러리에 포함되어 있습니다. 미리 컴파일 되어있는 정적 라이브러리의 코드를 호출해서 사용하기 때문에 컴파일 실행시간이 줄어들지만, 라이브러리 전체를 포함하고 있어 프로그램의 크기가 커질 수 있습니다.

문제 2

오른쪽 코드를 vim을 통해서 직접 입력하시고, add.c 를 만들고, 다음의 세가지 캡쳐본을 만드시오.

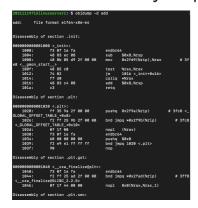
gcc -o 옵션을 통해 만들어진 실행 파일의 실행 결과

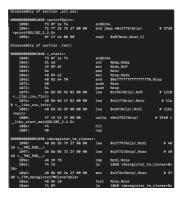
```
[2021111971@linuxserver1:~$ gcc add.c -o add [2021111971@linuxserver1:~$ ./add Sum : 8
```

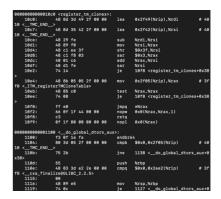
gcc -S 를 통해 만들어진 어셈블리

gcc -c 를 통해 만들어진 오브젝트 파일로부터 읽어들인 어셈블리어

활용 키워드: objdump -d







이하 생략..

문제 3

(gdb) break main (gdb) run (gdb) list (gdb) disassemble sumstore

(gdb) call add(1,7)

커멘드 분석

break main	(gdb) break main Breakpoint 1 at 0x1161: file add.c, line 9.
	add.c 파일의 9 번째 라인의 main 함수에 브레이크 포인트를
	설정합니다
	0x1161 은 main 함수가 실행을 시작하는 메모리의
	주소입니다.
run	<pre>(gdb) run Starting program: /home/2021111971/add Breakpoint 1, main () at add.c:9 9 int main() {</pre>
	/home/2021111971/add 에 위치한 프로그램을 실행합니다.
	break main 명령어로 설정한 브레이크 포인트에 중단합니다.
	9 번째 줄에서 중단되었고, 9 번째 줄을 출력합니다.

```
[(gdb) list
list
                                 5
                                             int add(int a, int b) {
                                 6
                                                  return a + b;
                                  7
                                 8
                                 9
                                             int main() {
                                 10
                                                  int sum = add(3, 5);
                                                  printf("Sum : %d\n", sum);
                                 11
                                 12
                                                  return 0;
                                 13
                                 main 함수를 기준으로 소스코드를 출력합니다
                                 10 줄 단위로 출력합니다
                                 (gdb) disassemble sumstore
disassemble sumstore
                                 No symbol "sumstore" in current context.
                                 (gdb) disassemble
Dump of assembler code for function main:
    → disassemble
                                    0x0000555555555161 <+0>:
0x00005555555555165 <+4>:
                                                           push %rbp
                                                                 %rsp,%rbp
$0x10,%rsp
                                    0x00005555555555166 <+5>:
                                                           mov
                                                                 $0x5,%esi
$0x3,%edi
                                    0x0000555555555516d <+12>:
                                                           mov
                                    0x00005555555555172 <+17>:
                                                           mov
                                                           callq
                                                                 %eax,-0x4(%rbp)
-0x4(%rbp),%eax
                                    0x00000555555555517c <+27>:
0x00005555555555517f <+30>:
                                                           mov
mov
                                                   <+33>:
                                                           mov
                                                                 %eax,%esi
                                                                                    # 0x55555556004
                                    0x0000555555555184 <+35>:
                                                           lea
                                                                 0xe79(%rip),%rdi
                                    0x000055555555518b <+42>:
                                                                 $0x0,%eax
                                                           mov
callq
                                                   <+47>:
                                    0x00005555555555195 <+52>:
                                                           mov
leaveq
                                                                 $0x0,%eax
                                       000555555555519a <+57>:
                                 End of assembler dump.
                                 main 함수에서 작동하는 기계 명령어를 표시합니다.
                                 시작주소 <사용메모리> 수행 내용으로 출력합니다.
                                 [(gdb) call add(1,7)
call add(1,7)
                                 $1 = 8
                                 1과 7을 매개변수로 add 함수를 호출합니다.
                                 함수의 반환 값을, $1 에 저장해 출력합니다.
```

문제 4

calc.o 파일 생성 후 내용 확인

```
2021111971@linuxserver1:~$ gcc -c calc.c
2021111971@linuxserver1:~$ objdump -d calc.o
                                       file format elf64-x86-64
calc.o:
Disassembly of section .text:
                                                                                                                                                                                                                     000000002e <main>:
f3 0f 1e fa
55
48 89 e5
48 83 ec 10
be 05 00 00 00
be 05 00 00 00
e8 00 00 00 00
89 45 f8
be 05 00 00 00
89 45 f5
88 00 00 00 00
89 45 f6
48 8d 3d 00 00 00
88 00 00 00
89 46 fc
89 c6
48 8d 3d 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00
88 00 00 00 00
88 00 00 00 00
88 00 00 00 00
88 00 00 00 00
88 00 00 00 00
0000000000000000000 <add>:
                         endbr64
                                                                                                                                                                                                      2e: 32: 36: 36: 36: 36: 56: 56: 56: 56: 56: 77: 79: 88: 88: 89:
                                                                                                                                                                                                                                                                                                          4
%rbp
%rsp,%rbp
$0x10,%rsp
$0x5,%esi
$0x3,%edi
49 <main+0x1b>
%eax,-0x8(%rbp)
$0x5,%esi
$0x3,%edi
                                                                                                                                  %rbp
                                                                                                                                                                                                                                                                                        push mov sub mov callq mov mov callq mov mov callq mov mov callq mov mov lea
                                                                                                                                %rsp,%rbp
%edi,-0x4(%rbp)
%esi,-0x8(%rbp)
-0x4(%rbp),%edx
-0x8(%rbp),%eax
         5:
8:
                                                                                                           mov
                                                                                                           mov
       b:
e:
11:
                                                                                                           mov
                                                                                                           mov
                          8b 45 f8
01 d0
5d
                                                                                                                                  %edx,%eax
        14:
                                                                                                           add
        16:
                                                                                                                                                                                                                                                                                                          pop
        17:
   000000000000018 <minus>:
                          f3 0f 1e fa
                                                                                                                                                                                                                                                                                                                                                                        # 6a <main+0x3c>
      18:
1c:
                                                                                                          endbr64
                                                                                                                                                                                                                                                                                       mov
callq
mov
mov
lea
mov
callq
                                                                                                          push
                                                                                                                                %rbp
                         48 89 e5
89 7d fc
89 75 f8
8b 45 fc
2b 45 f8
                                                                                                                                %rsp,%rbp
%rsp,%rbp
%edi,-0x4(%rbp)
%esi,-0x8(%rbp)
-0x4(%rbp),%eax
      1d:
20:
23:
26:
29:
                                                                                                           mov
                                                                                                          mov
                                                                                                           mov
                                                                                                                                                                                                                                                                                                                                                                        # 80 <main+0x52>
                                                                                                           mov
                                                                                                                                  -0x8(%rbp),%eax
                                                                                                           sub
                                                                                                                                                                                                                                                                                         mov
leaveq
retq
       2c:
2d:
                          5d
c3
                                                                                                                                  %rbp
```

함수의 하위 주소가 0으로 채워져 있음

실행파일 생성 후 내용확인

```
321111971@linuxserver1:~$ gcc calc.o -o calc
2021111971@linuxserver1:~$ objdump -d calc
calc:
          file format elf64-x86-64
Disassembly of section .init:
0000000000001000 <_init>:
    1000:
                f3 0f 1e fa
                                         endbr64
    1004:
                48 83 ec 08
                                         sub
                                                $0x8,%rsp
                48 8b 05 d9 2f 00 00
    1008:
                                                0x2fd9(%rip),%rax
                                         mov
                                                                          # 3fe8 <__gmon_st
art__>
100f:
                48 85 c0
                                         test
                                                %rax,%rax
    1012:
                74 02
                                         jе
                                                1016 <_init+0x16>
    1014:
                ff d0
                                         callq
                                                *%rax
    1016:
                48 83 c4 08
                                         add
                                                $0x8,%rsp
    101a:
                                         retq
Disassembly of section .plt:
```

```
0000000000001177 <main>:
                f3 0f 1e fa
    1177:
                                          endbr64
    117b:
                 55
                                          push
                                                 %rbp
                48 89 e5
    117c:
                                          mov
                                                 %rsp,%rbp
                                                 $0x10,%rsp
    117f:
                 48 83 ec 10
                                          sub
    1183:
                be 05 00 00 00
                                          mov
                                                 $0x5,%esi
    1188:
                bf 03 00 00 00
                                                 $0x3,%edi
                                          mov
    118d:
                e8 b7 ff ff
                                          callq
                                                 1149 <add>
                89 45 f8
                                                 %eax,-0x8(%rbp)
    1192:
                                          mov
                                                 $0x5,%esi
    1195:
                be 05 00 00 00
                                          mov
                bf 03 00 00 00
    119a:
                                          mov
                                                 $0x3,%edi
                                                 1161 <minus>
    119f:
                e8 bd ff ff ff
                                          callq
    11a4:
                 89 45 fc
                                          mov
                                                 %eax,-0x4(%rbp)
                                                 -0x8(%rbp),%eax
                8b 45 f8
    11a7:
                                          mov
    11aa:
                89 c6
                                                 %eax,%esi
                                          mov
    11ac:
                48 8d 3d 51 0e 00 00
                                                 0xe51(%rip),%rdi
                                                                          # 2004 <_IO_stdin_
                                          lea
used+0x4>
                b8 00 00 00 00
    11b3:
                                          mov
                                                 $0x0,%eax
    11b8:
                e8 93 fe ff ff
                                                 1050 <printf@plt>
                                          callq
                8b 45 fc
    11bd:
                                          mov
                                                 -0x4(%rbp),%eax
    11c0:
                89 c6
                                          mov
                                                 %eax,%esi
    11c2:
                 48 8d 3d 45 0e 00 00
                                          lea
                                                 0xe45(%rip),%rdi
                                                                          # 200e <_IO_stdin_
used+0xe>
    11c9:
                b8 00 00 00 00
                                          mov
                                                 $0x0,%eax
                 e8 7d fe ff ff
    11ce:
                                          callq
                                                 1050 <printf@plt>
                                                 $0x0,%eax
    11d3:
                b8 00 00 00 00
                                          mov
    11d8:
                 с9
                                          leaveq
                с3
    11d9:
                                          retq
    11da:
                66 Of 1f 44 00 00
                                          nopw
                                                 0x0(%rax,%rax,1)
```

함수의 하위비트에 값이 채워졌습니다.

링커의 역할

→ 오브젝트 파일에서는 정확한 함수주소가 결정되지 않습니다. 실행파일로 완전히 컴파일 될 때 링커가 함수의 최종 메모리 주소를 결정합니다.