



2025 SWING REVERSING

32기 리버싱 스터디



2025

SWING





담당자 노희민





리버싱 1주 스터디

진행 순서


 OT 내용 리마인드

 정적 분석/동적 분석

 컴퓨터 구조

 어셈블리 명령어

 팀 구성원 소개

 일정 안내



진행 순서를 알아봅니다.



Goals

어셈블리어 코드 해석에 필요한 기초 지식 숙지

Chapter.1

OT 내용 리마인드

Chapter.5

리버싱 실습

Chapter.2

정적 분석/동적 분석

Chapter.6

IDA Freeware


Chapter.3


컴퓨터 구조


Chapter.4


어셈블리 명령어


OT 리마인드

 미팅 진행 순서


 프로젝트 배경 소개

 목표 및 목적

 프로젝트 범위

 프로젝트 우선순위

 팀 구성원 소개

 일정 안내



OT 리마인드

리버싱이란?

시스템을 **역추적**하여 시스템의 구조를
알아내는 행위.

정적 분석

- 파일 실행 X
- 악성 프로그램 감염 위험 적음
- 프로그램 전체 구조 파악
- 동적 요소 고려 어려움

동적 분석

- 파일 실행 O
- 프로그램 동적 파악
- 분석 환경 구축 어려움

KICK-OFF MEETING

미팅 진행 순서

✓ 프로젝트 배경 소개

목표 및 목적

프로젝트 범위

프로젝트 우선순위

팀 구성원 소개

✓ 일정 안내

MIRI COMPANY



OT 리마인드 : 디스어셈블과 어셈블리어

프로그래밍 언어

고급 언어

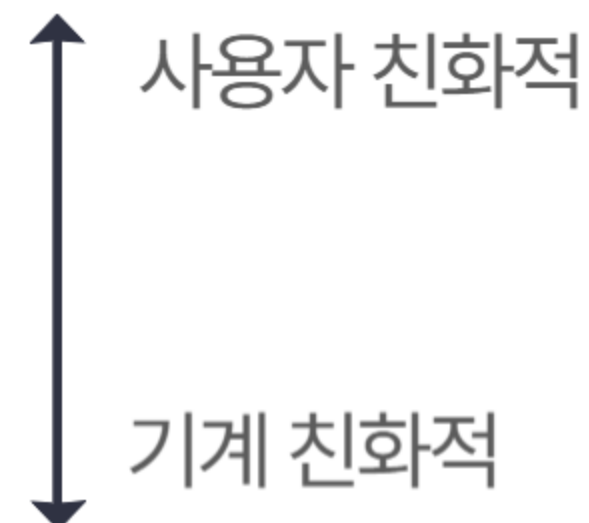
```
printf("hello world");
```

어셈블리어

```
LEA esi, edi
```

기계어(바이너리)

```
0111000101000000
```



디스어셈블

프로그램(기계어) -> 어셈블리어

컴파일

고급 언어 -> 기계어(프로그램)



KICK-OFF MEETING



미팅 진행 순서



프로젝트 배경 소개



목표 및 목적



프로젝트 범위



프로젝트 우선순위



팀 구성원 소개



일정 안내



OT 리마인드 : 어셈블리어

```
MOV EBP, ESP
```

????

이거... 어떻게 읽는 걸까?



KICK-OFF MEETING



미팅 진행 순서



프로젝트 배경 소개



목표 및 목적



프로젝트 범위



프로젝트 우선순위



팀 구성원 소개



일정 안내



OT 리마인드 : 어셈블리어 구조

MOV EBP, ESP

범용 레지스터

명령어 피연산자

피연산자

- 상수
- 레지스터
- 메모리



KICK-OFF MEETING



미팅 진행 순서



프로젝트 배경 소개



목표 및 목적



프로젝트 범위



프로젝트 우선순위



팀 구성원 소개



일정 안내

MIRI COMPANY



OT 리마인드 : 주요 명령어

명령 코드

데이터 이동(Data Transfer)

`mov`, `lea`

산술 연산(Arithmetic)

`inc`, `dec`, `add`, `sub`

논리 연산(Logical)

`and`, `or`, `xor`, `not`

비교(Comparison)

`cmp`, `test`

분기(Branch)

`jmp`, `je`, `jg`

스택(Stack)

`push`, `pop`

프로시저(Procedure)

`call`, `ret`, `leave`

시스템 콜(System call)

`syscall`

Q1. mov/lea 차이는?

Q2. inc, dec의 역할?



KICK-OFF MEETING



미팅 진행 순서



프로젝트 배경 소개



목표 및 목적



프로젝트 범위



프로젝트 우선순위



팀 구성원 소개



일정 안내



OT 리마인드 : 주요 명령어 (2) 논리 연산

AND

dst와 src의 비트가 모두 1이면 1

OR

dst와 src의 비트 중 하나라도 1이면 1

XOR

dst와 src의 비트가 서로 다르면 1

NOT

매개변수의 비트 전부 반전

KICK-OFF MEETING

미팅 진행 순서

✓ 프로젝트 배경 소개

목표 및 목적

프로젝트 범위

프로젝트 우선순위

팀 구성원 소개

✓ 일정 안내



OT 리마인드 : 주요 명령어 (2) 논리 연산

AND 연산 예제

```
1  [Register]
2  eax = 0xffff0000
3  ebx = 0xcafebabe
4
5  [Code]
6  and eax, ebx
7
8  [Result]
9  eax = 0xcafe0000
```

16진수 -> 2진수 변환
이후 비트 연산

KICK-OFF MEETING

미팅 진행 순서

✓ 프로젝트 배경 소개

목표 및 목적

프로젝트 범위

프로젝트 우선순위

팀 구성원 소개

✓ 일정 안내

MIRI COMPANY



OT 리마인드 : 주요 명령어 (2) 논리 연산

AND 연산

dst와 src의 비트가 모두 1이면 1

16진수

2진수

0xffff0000

1111 1111 1111 1111 0000 0000 0000 0000

0xcafebabe

1100 1010 1111 1110 1011 1010 1011 1110

0xcafe0000

1100 1010 1111 1110 0000 0000 0000 0000

KICK-OFF MEETING

- 미팅 진행 순서
- ✓ 프로젝트 배경 소개
- 목표 및 목적
- 📁 프로젝트 범위
- 📢 프로젝트 우선순위
- 👤 팀 구성원 소개
- ✓ 일정 안내

MIRI COMPANY



OT 리마인드 : 피연산자(1) 범용 레지스터

```
MOV EBP, ESP
```

이제 해석 가능!

MOV: 오퍼랜드 1에 오퍼랜드 2의 값을 복사하는 명령어.

ESP, EBP: 스택 프레임의 끝과 시작 주소 저장하는 레지스터.

스택 프레임을 만드는 함수구나!



스택 프레임이 뭐지?



KICK-OFF MEETING



미팅 진행 순서



프로젝트 배경 소개



목표 및 목적



프로젝트 범위



프로젝트 우선순위



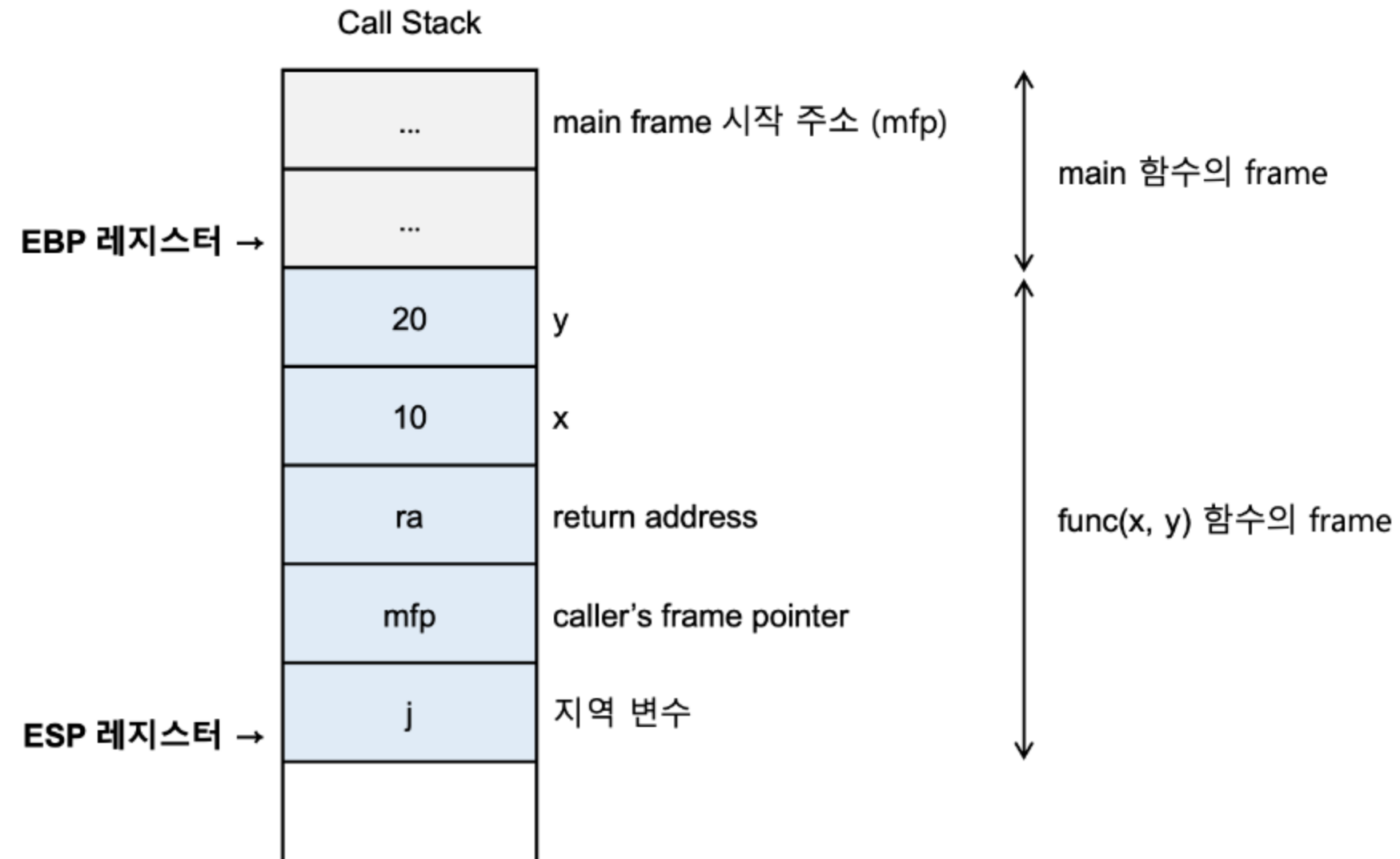
팀 구성원 소개



일정 안내



OT 리마인드 : 스택 프레임



스택 프레임: 각 스레드(함수)별로 할당되는 스택 영역.

KICK-OFF MEETING

미팅 진행 순서

✓ 프로젝트 배경 소개

목표 및 목적

프로젝트 범위

프로젝트 우선순위

팀 구성원 소개

✓ 일정 안내

MIRI COMPANY



OT 리마인드 : 스택 프레임

```
int main() {  
    int a = 0;  
    int b = 3;  
    return a + b;  
}
```

```
+-----+  
| Caller EBP |  
+-----+ <-- EBP  
| Return Address |  
+-----+  
| a = 0 | (EBP - 4)  
+-----+  
| b = 3 | (EBP - 8)  
+-----+ <-- ESP
```

KICK-OFF MEETING

미팅 진행 순서

✓ 프로젝트 배경 소개

목표 및 목적

프로젝트 범위

프로젝트 우선순위

팀 구성원 소개

✓ 일정 안내

MIRI COMPANY



OT 리마인드 : 스택 프레임

```
int main() {  
    int a = 0;  
    int b = 3;  
    return a + b;  
}
```

Q3 : 해당 함수가 종료되고 나면
스택 프레임은 어떻게 변할까?

```
+-----+  
| Caller EBP |  
+-----+ <-- EBP  
| Return Address |  
+-----+  
| a = 0          | (EBP - 4)  
+-----+  
| b = 3          | (EBP - 8)  
+-----+ <-- ESP
```

KICK-OFF MEETING

-  미팅 진행 순서
-  프로젝트 배경 소개
-  목표 및 목적
-  **프로젝트 범위**
-  프로젝트 우선순위
-  팀 구성원 소개
-  일정 안내



실전으로 : IDA Freeware

이제 어셈블리어도 읽을 수 있고, 스택 프레임도 뭔지 알겠어!!
실제 프로그램을 리버싱 해보자!!



무슨 툴로?

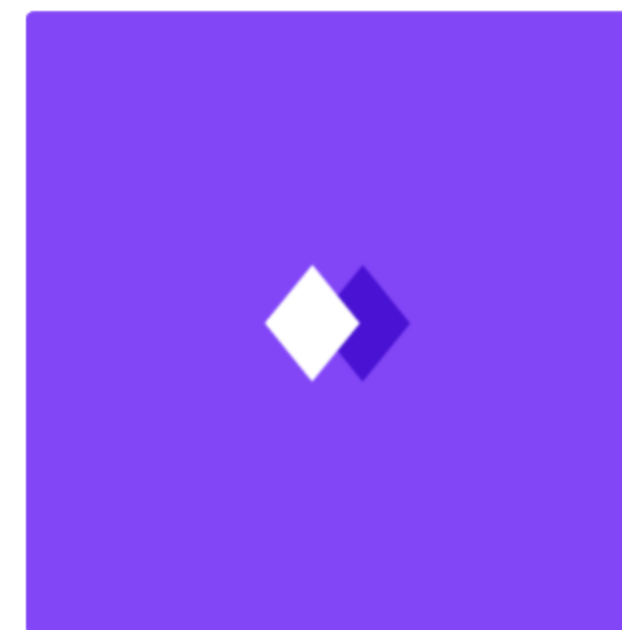
KICK-OFF MEETING


- 미팅 진행 순서
- ✓ 프로젝트 배경 소개
- 목표 및 목적
- 프로젝트 범위
- 프로젝트 우선순위
- 팀 구성원 소개
- ✓ 일정 안내

MIRI COMPANY



실전으로 : IDA Freeware



Exercise: Helloworld  ⋮

9.7★ (417)

보유 중

강의 시작

상세정보

⌚ 약 1시간 30분 소요

🔊 쉬운 난이도

☑ IDA



KICK-OFF MEETING



미팅 진행 순서



프로젝트 배경 소개



목표 및 목적



프로젝트 범위



프로젝트 우선순위



팀 구성원 소개



일정 안내



IDA 설치 방법



Free and still mighty ↩

See IDA in action and get to know the most powerful disassembler and decompiler at no cost.

What do you get with IDA Free?

- Support for x86/x86-64bit processors and 32-bit/64-bit applications
- x86/x86-64bit cloud-based decompiler
- Save your analysis

[Check documentation](#) ↗

[Download IDA Free](#) →

Free 설치 및 라이선스 등록 함께 진행
(이미 설치되어 있을 경우 패스)



KICK-OFF MEETING



미팅 진행 순서



프로젝트 배경 소개



목표 및 목적



프로젝트 범위



프로젝트 우선순위



팀 구성원 소개



일정 안내



과제

1. 16진수 <-> 2진수 변환 문서화
 - a. 1 ~ 50 중 랜덤 숫자 하나 골라서 16진수 변환
 - b. 이후 해당 16진수 2진수 변환
 - c. 비트 반전 결과 16진수로 재변환
 2. 슬라이드 Q1,2,3 풀이. (단답형)
 3. swinghi.exe 정적 분석 실습
 - a. 디어셈블리 및 분석 문서화
 - b. hello() 어셈블리 줄별 해석
 - c. main() 함수 스택 프레임 변화 그려서 첨부
- > 카페에 과제 가이드라인 및 문제 파일 업로드



수고하셨습니다~

질문사항은 카카오톡 주세요



2025

SWING



담당자 노희민