



Libasm

Assembly yourself!

*Summary: The aim of this project is to get familiar with assembly language.*

**요약: 이 프로젝트의 목표는 어셈블리어에 친숙해지는 것입니다.**

# Contents

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>Common Instructions</b>	<b>3</b>
<b>III</b>	<b>Mandatory part</b>	<b>4</b>
<b>IV</b>	<b>Bonus part</b>	<b>5</b>

# Chapter I

## Introduction

An assembly (or assembler) language, often abbreviated asm, is a low-level programming language for a computer, or other programmable device, in which there is a very strong (but often not one-to-one) correspondence between the language and the architecture's machine code instructions. Each assembly language is specific to a particular computer architecture. In contrast, most high-level programming languages are generally portable across multiple architectures but require interpreting or compiling. Assembly language may also be called symbolic machine code.

종종 asm으로 축약되는 어셈블리(또는 어셈블러)언어는 저수준 프로그래밍 언어입니다.  
이 언어는 컴퓨터 또는 다른 프로그래밍이 가능한 장치를 위한 언어입니다.  
이 언어는 언어와 아키텍처의 기계어 코드 지시자의 대응이 매우 긴밀합니다(종종 일대일은 아님).

각 어셈블리 언어는 특정 컴퓨터 아키텍처에 따라 특정됩니다.  
반대로, 대부분의 고수준 프로그래밍 언어들은 일반적으로 여러 아키텍처에 걸쳐 이식이 가능하지만 인터프리팅 또는 컴파일이 필요합니다.  
어셈블리어는 심볼릭 머신 코드라고도 합니다.

# Chapter II

## Common Instructions

당신의 함수들은 UB와는 별개로, 갑자기 종료되면 안됩니다.( 세그폴트, 버스 에러, 더블 프리 등)  
만약에 그러면, 기능적으로 동작하지 않는 걸로 간주하고 0점 받으실거예요.

- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and you will receive a 0 during the evaluation.
- Your Makefile must at least contain the rules `$(NAME)`, `all`, `clean`, `fclean` and `re`. And must recompile/relink only necessary files.
- To turn in `bonuses` to your project, you `must` include a `rule bonus` to your Makefile, which will add all the various headers, libraries or functions that are forbidden on the main part of the project. Bonuses `must` be in a `different file _bonus.{c/h}`. Mandatory and bonus part evaluation is done separately.
- We encourage you to create test programs for your project even though this work **won't have to be submitted and won't be graded**. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.
- You must write `64 bits` ASM. Beware of the `"calling convention"`. "호출 규약"
- You `can't do inline` ASM, you must do `'s'` files.
- You must compile your assembly code with `nasm`.
- You `must use` the `Intel syntax`, not the AT&T.

# Chapter III

## Mandatory part

- The library must be called `libasm.a`.
- You must submit a `main` that will test your functions and that will compile with your library to show that it's functional.
- You must rewrite the following functions in asm:
  - `ft_strlen` (man 3 strlen)
  - `ft_strcpy` (man 3 strcpy)
  - `ft_strcmp` (man 3 strcmp)
  - `ft_write` (man 2 write)
  - `ft_read` (man 2 read)
  - `ft_strdup` (man 3 strdup, you can call to malloc)
- You must check for errors during syscalls and properly set them when needed
- Your code must set the variable `errno` properly.
- For that, you are allowed to call the `extern __error`.

# Chapter IV

## Bonus part

You can rewrite these functions in asm. The linked list function will use the following structure:

```
typedef struct          s_list
{
    void                *data;
    struct s_list       *next;
}                       t_list;
```

- `ft_atoi_base` (like the one in the piscine)
- `ft_list_push_front` (like the one in the piscine)
- `ft_list_size` (like the one in the piscine)
- `ft_list_sort` (like the one in the piscine)
- `ft_list_remove_if` (like the one in the piscine)