



## Philosophers

*철학이 이렇게 치명적일 줄은 생각도 못 했다.*

*요약: 이 프로젝트에서, 프로세스 스레딩의 기본과 동일한 메모리 공간에서 작업하는 방법을 배우게 됩니다.  
스레드를 만드는 방법을 배우게 될 것입니다. 뮤텍스, 세마포어, 공유 메모리를 발견하게 될 것입니다.*

- \* 이 파일은 프로젝트 수행을 좀 더 빠르고 편하게 하기 위해 만들어진 파일입니다.
- \* 개인적으로 보기위해 만들어진 파일이므로 오역이 있을 수 있습니다.
- \* 영어버전과 함께 보시는 걸 추천 드립니다.

# Contents

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>II</b>	<b>Mandatory part</b>	<b>3</b>

# Chapter I

## Introduction

철학(그리스어, 철학, 문자 그대로 "지혜의 사랑")은 존재, 지식, 가치관, 이성, 마음, 언어에 관한 일반적이고 근본적인 문제들에 대한 학문이다. 그러한 문제들은 종종 연구하거나 해결해야 할 문제로 제기된다. 이 용어는 아마도 피타고라스 (기원전 570년 - 495년)에 만들어졌을 것이다. 철학적인 방법으로는 질문, 비판적 토론, 이성적인 논쟁, 체계적인 표현 등이 있다. 고전적인 철학적 질문은 다음과 같다. 무엇이든 알고 또 그것을 증명하는 것이 가능한가? 무엇이 정말 진짜인가? 철학자들은 또한 다음과 같은 좀 더 실질적이고 구체적인 질문을 제기한다. 살기 가장 좋은 방법이 있을까? 정의로운 것이 더 나은가 아니면 불공평한 것이 더 나은가(만약 빠져나갈 수 있다면)? 인간에게는 자유가 있는가?

역사적으로, "철학"은 지식의 모든 부분을 포괄했다. 고대 그리스 철학자 아리스토텔레스 시대부터 19세기까지 "자연 철학"은 천문학, 의학, 물리학을 포괄했다. 예를 들어, 뉴턴의 1687년 자연 철학의 수학적 원리는 후에 물리학의 책으로 분류되었다. 19세기에는, 현대 연구 대학의 성장이 학문 철학과 다른 학문들을 전문화하고 전문화하도록 이끌었다. 현대에는 전통적으로 철학의 일부였던 조사들이 심리학, 사회학, 언어학, 경제학 등 별개의 학문적 학문들이 되었다.

예술, 과학, 정치, 또는 다른 추적과 밀접하게 관련된 다른 조사들은 철학의 일부로 남아 있었다. 예를 들어, 아름다움은 객관적일까 주관적일까? 과학적인 방법이 많은가 아니면 단지 하나뿐인가? 정치적인 유토피아는 희망적인 꿈일까 아니면 절망적인 환상일까? 학문 철학의 주요 하위 분야로는 형이상학("현실과 존재의 근본적 본질과 관계된"), 인식론학("지식 [및]...그 한계와 타당성"에 관한 것), 윤리학, 미학, 정치철학, 논리학, 과학철학이 있다.

# Chapter II

## Mandatory part

당신은 3개의 다른 프로그램을 써야 하지만 그들은 같은 기본 규칙을 가질 것입니다.

- 많은 철학자들은 둥근 테이블에 앉아 세 가지 중 하나를 하고 있다: 먹는 것, 생각하는 것, 자는 것.
- 먹으면서 생각하거나 자는 것이 아니고, 자는 동안에는 먹거나 생각하지 않으며, 물론 생각하면서 먹거나 하지 않는다.
- 철학자들은 커다란 스파게티 그릇을 중앙에 두고 원형 테이블에 앉아 있다.
- 탁자 위에 포크가 몇 개 있다.
- 스파게티는 포크 하나로 잡아서 먹기가 어렵기 때문에, 철학자는 포크를 한 손에 하나씩 들고 먹어야 한다고 가정합니다.
- 철학자들은 절대 굶어 죽지 말아야 한다.
- 모든 철학자는 먹어야 한다.
- 철학자들은 서로 말을 하지 않는다.
- 철학자들은 또 다른 철학자가 언제 죽을지 모른다.
- 철학자는 밥을 다 먹을 때마다 포크를 내려놓고 잠을 자기 시작할 것이다.
- 철학자는 잠을 다 자면 생각하기 시작할 것이다.
- 그 시뮬레이션은 철학자가 죽으면 멈춘다.
- 각 프로그램에는 동일한 옵션이 있어야 한다: `number_of_philosophers` `time_to_die` `time_to_eat` `time_to_sleep` [`number_of_times_each_philosopher_must_eat`]
  - `number_of_philosophers`: 철학자의 수와 포크의 수이다.
  - `time_to_die`: 밀리 초 단위이며, 만약 철학자가 마지막 식사를 시작하거나 시뮬레이션을 시작한 후 '`time_to_die`' milliseconds를 먹기 시작하지 않는다면 죽는다.
  - `time_to_eat`: 밀리 초 단위이며, 철학자가 식사하는 데 걸리는 시간이다. 그 시간 동안 그는 포크 두개를 유지해야 할 것이다.

- `time_to_sleep`: 밀리 초 단위이며, 철학자가 잠을 자는 데 쓰는 시간이다.
- `number_of_times_each_philosopher_must_eat`: 논쟁은 선택사항이며, 만약 모든 철학자들이 적어도 '`number_of_times_each_philosopher_must_eat`'을 먹는다면, 시뮬레이션은 중단될 것이다. 구체적으로 명시하지 않으면, 시뮬레이션은 철학자가 사망 할 때만 중단될 것입니다.

- 각 철학자에게는 1부터 '`number_of_philosophers`'까지의 숫자를 부여해야 한다.
- 철학자 숫자 1은 철학자 숫자 '`number_of_philosophers`' 옆에 있다. 숫자 N을 가진 다른 철학자는 철학자 N - 1과 철학자 N + 1 사이에 앉는다.
- 철학자의 상태 변화는 다음과 같이 기록되어야 한다. (X를 철학자 숫자로 `timestamp_in_ms`를 현재 타임스탬프로 대체. 단위는 밀리초)
  - `timestamp_in_ms` X가 포크를 잡았다.
  - `timestamp_in_ms` X가 먹는 중이다.
  - `timestamp_in_ms` X가 자는 중이다.
  - `timestamp_in_ms` X가 생각하는 중이다.
  - `timestamp_in_ms` X 죽었다.
- 출력된 상태는 다른 철학자의 상태와 뒤섞이거나 얹혀서는 안된다.
- 철학자의 죽음과 그 죽음을 출력할 때의 사이에 시간 10 ms를 초과할 수 없다.
- 다시 말하지만, 철학자들은 죽는 것을 피해야 한다!

<b>Program name</b>	<code>philo_one</code>
<b>Turn in files</b>	<code>philo_one/</code>
<b>Makefile</b>	Yes
<b>Arguments</b>	<code>number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]</code>
<b>External functs.</b>	<code>malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock, pthread_mutex_unlock, memset</code>
<b>Libft authorized</b>	No
<b>Description</b>	스레드와 뮤텍스를 가진 철학자

이 버전에서 일반적이지 않은 규칙은 다음과 같다:

- 각 철학자들 사이에는 하나의 포크가 있으므로, 각 철학자의 오른쪽과 왼쪽에는 포크가 있을 것이다.

- 철학자들이 포크를 복제하는 것을 피하기위해, 각각의 포크에 뮤텍스로 포크 상태를 보호해야한다.
- 각 철학자는 스레드이어야 한다.

<b>Program name</b>	philo_two
<b>Turn in files</b>	philo_two/
<b>Makefile</b>	Yes
<b>Arguments</b>	number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]
<b>External functs.</b>	malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, sem_open, sem_close, sem_post, sem_wait, sem_unlink
<b>Libft authorized</b>	No
<b>Description</b>	스레드와 세마포어를 가진 철학자

이 버전에서 일반적이지 않은 규칙은 다음과 같다:

- 모든 포크는 테이블 가운데에 있다.
- 그들은 메모리에는 상태가 없지만, 사용 가능한 포크 수는 세마포어로 표현된다.
- 각 철학자는 스레드이어야 한다.

<b>Program name</b>	philo_three
<b>Turn in files</b>	philo_three/
<b>Makefile</b>	Yes
<b>Arguments</b>	number_of_philosophers time_to_die time_to_eat time_to_sleep [number_of_times_each_philosopher_must_eat]
<b>External functs.</b>	malloc, free, write, fork, kill, exit, pthread_create, pthread_detach, pthread_join, usleep, gettimeofday, waitpid, sem_open, sem_close, sem_post, sem_wait, sem_unlink
<b>Libft authorized</b>	No
<b>Description</b>	프로세스와 세마포어를 가진 철학자

이 버전에서 일반적이지 않은 규칙은 다음과 같다:

- 모든 포크는 테이블 가운데에 있다.
- 그들은 메모리에는 상태가 없지만, 사용 가능한 포크 수는 세마포어로 표현된다.
- 각 철학자는 하나의 프로세스이어야 하고 주 프로세스는 철학자가 되어서는 안된다.