



Filósofos

Nunca pensei que a filosofia fosse tão mortal

Resumo:

*Neste projeto, você aprenderá os fundamentos da segmentação de um processo.
Você verá como criar threads e descobrirá mutexes.*

Versão: 11

Conteúdo

EU	Introdução	2
II	Instruções Comuns	3
III	Visão geral	5
4	Regras globais	6
V	Parte obrigatória	8
VI	parte bônus	9
VII	Submissão e avaliação por pares	10

Capítulo I

Introdução

Filosofia (do grego, *philosophia*, literalmente "amor à sabedoria") é o estudo de assuntos gerais e questões fundamentais sobre existência, conhecimento, valores, razão, mente e linguagem. Tais questões são frequentemente colocadas como problemas a serem estudados ou resolvidos. O termo foi provavelmente cunhado por Pitágoras (c. 570 – 495 aC). Os métodos filosóficos incluem questionamento, discussão crítica, argumento racional e apresentação sistemática.

As questões filosóficas clássicas incluem: É possível saber alguma coisa e provar isto? O que é mais real? Os filósofos também colocam questões mais práticas e concretas, tais como: Existe uma melhor maneira de viver? É melhor ser justo ou injusto (se alguém puder escapar impune isto)? Os humanos têm livre arbítrio?

Historicamente, a "filosofia" abrangia qualquer corpo de conhecimento. Desde a época do filósofo grego antigo Aristóteles até o século XIX, a "filosofia natural" abrangia astronomia, medicina e física. Por exemplo, os *Princípios Matemáticos de Newton* de 1687 de Filosofia Natural mais tarde foi classificado como um livro de física.

No século XIX, o crescimento das modernas universidades de investigação levou a filosofia acadêmica e outras disciplinas a profissionalizarem-se e a especializarem-se. Na era moderna, alguns investigações que tradicionalmente faziam parte da filosofia tornaram-se disciplinas acadêmicas separadas, incluindo psicologia, sociologia, linguística e economia.

Outras investigações intimamente relacionadas com arte, ciência, política ou outras atividades permaneceram parte da filosofia. Por exemplo, a beleza é objetiva ou subjetiva? Existem muitos métodos científicos ou apenas um? A utopia política é um sonho esperançoso ou uma fantasia sem esperança? Os principais subcampos da filosofia acadêmica incluem a metafísica ("preocupada com a natureza fundamental da realidade e do ser"), a epistemologia (sobre a "natureza e os fundamentos da conhecimento [e]... seus limites e validade"), ética, estética, filosofia política, lógica e filosofia da ciência.

Capítulo II

Instruções Comuns

- Seu projeto deve ser escrito em C.
- Seu projeto deverá ser escrito de acordo com a Norma. Se você tiver arquivos/funções bônus, eles serão incluídos na verificação de norma e você receberá 0 se houver um erro de norma.
- Suas funções não devem encerrar inesperadamente (falha de segmentação, erro de barramento, liberação dupla, etc.) além de comportamentos indefinidos. Caso isso aconteça, seu projeto será considerado não funcional e receberá nota 0 na avaliação.
- Todo o espaço de memória alocado no heap deve ser liberado adequadamente quando necessário. Sem vazamentos será tolerado.
- Se o assunto exigir, você deve enviar um Makefile que irá compilar seus arquivos fonte para a saída necessária com as flags -Wall, -Wextra e -Werror, use cc, e seu Makefile não deve relinkar.
- Seu Makefile deve conter pelo menos as regras \$(NAME), all, clean, fclean e ré.
- Para entregar bônus ao seu projeto, você deve incluir uma regra bônus em seu Makefile, que adicionará todos os diversos cabeçalhos, bibliotecas ou funções que são proibidas na parte principal do projeto. Os bônus devem estar em um arquivo diferente _bonus.{c/h} se o assunto não especificar mais nada. A avaliação da parte obrigatória e da parte bônus é feita separadamente.
- Se o seu projeto permitir que você use sua libft, você deve copiar seus fontes e seu Makefile associado em uma pasta libft com seu Makefile associado. O Makefile do seu projeto deve compilar a biblioteca usando seu Makefile e depois compilar o projeto.
- Nós encorajamos você a criar programas de teste para o seu projeto, mesmo que este trabalho **não precise ser enviado e não receba notas**. Isso lhe dará a chance de testar facilmente seu trabalho e o de seus colegas. Você achará esses testes especialmente úteis durante sua defesa. Na verdade, durante a defesa, você é livre para usar seus testes e/ou os testes do colega que está avaliando.
- Envie seu trabalho para o repositório git designado. Somente o trabalho no repositório git será avaliado. Se Deepthought for designado para avaliar seu trabalho, isso será feito

após suas avaliações por pares. Se ocorrer um erro em qualquer seção do seu trabalho durante a avaliação do Deepthought, a avaliação será interrompida.

Capítulo III

Visão geral

Aqui estão as coisas que você precisa saber se quiser ter sucesso nesta tarefa:

- Um ou mais filósofos sentam-se numa mesa redonda.
Há uma tigela grande de espaguete no meio da mesa.
- Os filósofos alternativamente **comem, pensam** ou **dormem**.
Enquanto comem, não pensam nem dormem; enquanto pensam, não comem nem dormem; e, claro, enquanto dormem, não comem nem pensam.
- Também há garfos na mesa. Existem **tantos garfos quanto filósofos**.
- Como servir e comer espaguete com apenas um garfo é muito inconveniente, um filósofo pega o garfo direito e o esquerdo para comer, um em cada mão.
- Quando um filósofo termina de comer, ele coloca os garfos de volta na mesa e começa a dormir.
Uma vez acordados, eles começam a pensar novamente. A simulação termina quando um filósofo morre de fome.
- Todo filósofo precisa comer e nunca deveria passar fome.
- Os filósofos não falam uns com os outros.
- Os filósofos não sabem se outro filósofo está prestes a morrer.
- Não há necessidade de dizer que os filósofos deveriam evitar morrer!

Capítulo IV

Regras globais

Você tem que escrever um programa para a parte obrigatória e outro para a parte bônus (se você decidir fazer a parte bônus). Ambos devem cumprir as seguintes regras:

- Variáveis globais são proibidas!
- Seu(s) programa(s) deve(m) receber os seguintes argumentos:
número_de_filósofos hora_de_morrer hora_de_comer hora_de_dormir
[número_de_vezes_cada_filósofo_deve_comer]
 - `numero_de_filósofos`: O número de filósofos e também o número de garfos.
 - `time_to_die` (em milissegundos): Se um filósofo não começar a comer `time_to_die` milissegundos desde o início de sua última refeição ou do início da simulação, ele morre.
 - `time_to_eat` (em milissegundos): O tempo que um filósofo leva para comer. Durante esse tempo, eles precisarão segurar dois garfos.
 - `time_to_sleep` (em milissegundos): O tempo que um filósofo passará dormindo.
 - `number_of_times_each_philosopher_must_eat` (argumento opcional): se todos os filósofos comeram pelo menos `number_of_times_each_philosopher_must_eat` vezes, a simulação para. Se não for especificado, a simulação para quando um filósofo morre.
- Cada filósofo possui um número que varia de 1 a `numero_de_filósofos`.
- O filósofo número 1 fica ao lado do filósofo número `numero_de_filósofos`. Qualquer outro número de filósofo N fica entre o número de filósofo N - 1 e o número de filósofo N + 1.

Sobre os logs do seu programa:

- Qualquer mudança de estado de um filósofo deve ser formatada da seguinte forma:

ÿ timestamp_in_ms X tomou uma bifurcação

ÿ timestamp_in_ms X está comendo

ÿ timestamp_in_ms X está dormindo

ÿ timestamp_in_ms X está pensando

ÿ timestamp_in_ms X morreu

Substitua timestamp_in_ms pelo carimbo de data/hora atual em milissegundos e X pelo número do filósofo.

- Uma mensagem de estado exibida não deve ser confundida com outra mensagem.
- Uma mensagem anunciando a morte de um filósofo não deve ser exibida mais de 10 ms após a morte real do filósofo.
- Mais uma vez, os filósofos deveriam evitar morrer!



Seu programa não deve ter nenhuma **corrida de dados**.

Capítulo V

Parte obrigatória

Nome do programa	Filó
Entregar arquivos	Makefile, *.h, *.c, no diretório philo/
Makefile	NOME, tudo, limpo, fclean, re
Argumentos	número_de_filósofos hora_de_morrer hora_de_comer hora de dormir [número_de_vezes_each_philosopher_must_eat]
Funções externas.	memset, printf, malloc, grátis, escrever, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock, pthread_mutex_unlock
Libft autorizado	Não
Descrição	Filósofos com threads e mutexes

As regras específicas da parte obrigatória são:

- Cada filósofo deveria ser um fio.
- Existe uma bifurcação entre cada par de filósofos. Portanto, se houver vários filósofos, cada filósofo tem um garfo no lado esquerdo e um garfo no lado direito lado. Se houvesse apenas um filósofo, deveria haver apenas um garfo na mesa.
- Para evitar que os filósofos dupliquem bifurcações, você deve proteger o estado das bifurcações com um mutex para cada um deles.

Capítulo VI

Parte bônus

Nome do programa	filo_bonus
Entregar arquivos	Makefile, *.h, *.c, no diretório philo_bonus/
Makefile	NOME, tudo, limpo, fclean, re
Argumentos	número_de_filósofos hora_de_morrer hora_de_comer hora de dormir [número_de_vezes_each_philosopher_must_eat]
Funções externas.	memset, printf, malloc, grátis, escrever, fork, matar, saída, pthread_create, pthread_detach, pthread_join, usleep, gettimeofday, waitpid, sem_open, sem_close, sem_post, sem_wait, sem_unlink
Libft autorizado	Não
Descrição	Filósofos com processos e semáforos

O programa da parte bônus segue os mesmos argumentos do programa obrigatório. Tem que cumprir os requisitos do capítulo *de regras globais*.

As regras específicas para a parte bônus são:

- Todos os garfos são colocados no meio da mesa.
- Eles não possuem estados na memória, mas o número de bifurcações disponíveis é representado por um semáforo.
- Cada filósofo deveria ser um processo. Mas o processo principal não deve ser um filósofo.



A parte bônus só será avaliada se a parte obrigatória for PERFEITO. Perfeito significa que a parte obrigatória foi feita integralmente e funciona sem mau funcionamento. Se você não passou em TODOS os requisitos obrigatórios, sua parte do bônus não será avaliada de forma alguma.

Capítulo VII

Envio e avaliação por pares

Entregue sua tarefa em seu repositório Git normalmente. Somente os trabalhos dentro do seu repositório serão avaliados durante a defesa. Não hesite em verificar os nomes dos seus arquivos para garantir que estejam corretos.

Diretório de peças obrigatórias: philo/

Diretório da parte bônus: philo_bonus/

