Filósofos

Nunca pensei que a filosofia fosse tão mortal

Resumo:

Neste projeto, você aprenderá os conceitos básicos de encadeamento de um processo.
Você verá como criar threads e descobrirá mutexes.

Versão: 11

Conteúdo

Capítulo I

Introdução

Filosofia (do grego, *philosophia,* literalmente "amor à sabedoria") é o estudo da ciência geral.
e questões fundamentais sobre a 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 a.C.). Os métodos filosóficos incluem
questionamento, discussão crítica, argumentação 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, como
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, "filosofia" abrangia qualquer corpo de conhecimento. Desde o tempo do filósofo grego
antigo Aristóteles até o século XIX, "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 universidades de pesquisa modernas levou a filosofia acadêmica e
outras disciplinas a se profissionalizarem e se especializarem. Na era moderna, algumas
investigações que tradicionalmente faziam parte da filosofia tornaram-se disciplinas acadêmicas separadas,
incluindo psicologia, sociologia, linguística e economia.

Outras investigações intimamente relacionadas com a arte, a ciência, a política ou outras actividades 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 deve ser escrito de acordo com a Norma. Se você tiver arquivos/funções de bônus, eles serão incluídos na verificação da norma e você receberá um 0 se houver um erro de norma dentro.
- Suas funções não devem sair inesperadamente (falha de segmentação, erro de barramento, double free, etc.) além de comportamentos indefinidos. Se isso acontecer, seu projeto será considerado não funcional e receberá um 0 durante a 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 compilará seus arquivos de origem para a saída necessária com os sinalizadores -Wall, -Wextra e -Werror, use cc e seu Makefile não deve ser vinculado novamente.
- Seu Makefile deve conter pelo menos as regras $(NAME), all, clean, fclean e ré.
- Para entregar bônus ao seu projeto, você deve incluir um bônus de regra ao seu Makefile, que adicionará todos os vários 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 nada mais. A avaliação da parte obrigatória e bônus é feita separadamente.
- Se seu projeto permitir que você use sua libft, você deve copiar suas 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, em seguida, compilar o projeto.
- Nós o encorajamos a criar programas de teste para seu projeto, mesmo que esse trabalho **não precise ser enviado e não seja classificado.** Isso lhe dará uma chance de testar facilmente seu trabalho e o trabalho de seus colegas. Você achará esses testes especialmente úteis durante sua defesa. De fato, 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 classificado. Se o Deepthought for designado para classificar seu trabalho, isso será feito

depois de suas avaliações por pares. Se um erro acontecer em qualquer seção do seu trabalho durante a classificação 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 em uma mesa redonda.
  Há uma grande tigela de espaguete no meio da mesa.
- Os filósofos alternadamente **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. Há **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 seus garfos de volta na mesa e começa a dormir. Uma vez acordado, ele começa a pensar novamente. A simulação para quando um filósofo morre de fome.
- Todo filósofo precisa comer e nunca deve passar fome.
- Os filósofos não falam entre si.
- Os filósofos não sabem se outro filósofo está prestes a morrer.
- Não é preciso dizer que os filósofos devem evitar a morte!

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 têm que obedecer às seguintes regras:

- Variáveis globais são proibidas!

- Seu(s) programa(s) deve(m) aceitar os seguintes argumentos:
  número_de_filósofos tempo_para_morrer tempo_para_comer tempo_para_dormir
  [número_de_vezes_cada_filósofo_deve_comer]

  ÿ número_de_filósofos: O número de filósofos e também o número de garfos.

  ÿ 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.

  ÿ time_to_eat (em milissegundos): O tempo que um filósofo leva para comer. Durante esse tempo, eles precisarão segurar dois garfos.

  ÿ tempo_para_dormir (em milissegundos): O tempo que um filósofo passará dormindo.

  ÿ número_de_vezes_que_cada_filósofo_deve_comer (argumento opcional): Se todos 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 tem um número que varia de 1 a número_de_filósofos.

- O filósofo número 1 fica ao lado do filósofo número number_of_philosophers.
  Qualquer outro número filosófico N fica entre o número filosófico N - 1 e o número filosófico 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 sofreu 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 registro de data e 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 deve ser exibida no máximo 10 ms após a morte real do filósofo.

• Mais uma vez, os filósofos devem evitar morrer!

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

Capítulo V

Parte obrigatória

Nome do programa	Filosofia
Entregar arquivos	Makefile, *.h, *.c, no diretório philo/
Faça o arquivo	NOME, tudo, limpo, fclean, re
Argumentos	número_de_filósofos tempo_de_morrer tempo_de_comer hora_de_dormir [número_de_vezes_que_cada_filósofo_deve_comer]
Funções externas.	memset, printf, malloc, livre, escrever, usardormir, obterhoradodia, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_bloqueio, pthread_mutex_desbloqueio
Autorizado pela Libft	Não
Descrição	Filósofos com threads e mutexes

As regras específicas para a parte obrigatória são:

- Cada filósofo deve ser um fio condutor.
- 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 há apenas um filósofo, deve haver apenas um garfo na mesa.
- Para evitar que os filósofos dupliquem os garfos, você deve proteger o estado dos garfos com um mutex para cada um deles.

Capítulo VI

Parte bônus

Nome do programa	filo_bônus
Entregar arquivos	Makefile, *.h, *.c, no diretório philo_bonus/
Faça o arquivo	NOME, tudo, limpo, fclean, re
Argumentos	número_de_filósofos tempo_de_morrer tempo_de_comer hora_de_dormir [número_de_vezes_que_cada_filósofo_deve_comer]
Funções externas.	memset, printf, malloc, livre, escrever, bifurcar, matar, sair, pthread_create, pthread_detach, pthread_join, dormir, obter hora do dia, esperar pid, sem_open, sem_close, sem_post, sem_wait, sem_unlink
Autorizado pela Libft	Não
Descrição	Filósofos com processos e semáforos

O programa da parte bônus adota os mesmos argumentos do programa obrigatório.
Ele deve estar em conformidade com os requisitos do capítulo *Regras globais* .

As regras específicas para a parte de bônus são:

- Todos os garfos são colocados no meio da mesa.
- Eles não têm estados na memória, mas o número de garfos disponíveis é representado por um semáforo.
- Cada filósofo deve 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 integralmente feita e funciona sem problemas. Se você não passou em TODOS os requisitos obrigatórios, sua parte de bônus não será avaliada.

Capítulo VII

Submissão e avaliação por pares

Entregue sua tarefa no seu repositório Git como de costume. Somente o trabalho dentro do seu repositório será avaliado durante a defesa. Não hesite em verificar novamente os nomes dos seus arquivos para garantir que estejam corretos.

Diretório de partes obrigatórias: philo/

Diretório de bônus: philo_bonus/