

Atividade Prática

Simulador básico de um Gerenciador de processos usando Heap.

Prof. Valdigleis S Costa,
23 de junho de 2025.

Introdução.


Este trabalho prático visa aprofundar a compreensão dos conceitos fundamentais relacionados à estrutura de dados chamada Heap, em especial, nesta atividade deverá ser implementada uma Min-Heap. O Min-Heap será usado como estrutura de dados básica para ser usado na implementação de um pequeno (e simples) gerenciador de processos.

Para a realização deste trabalho, os discentes deverão utilizar os arquivos básicos (myProcess.h, myProcess.c, heap.h, heap.c e ManagerProcess.c) disponibilizados pelo professor. Sendo que tais documentos são descritos a seguir.

- myProcess.h: apresenta as estruturas básicas para descrever os processos (Process) e a fila (Queue) de processos.
- myProcess.c: contém a implementação das funções básicas para manipular os processos (Process) e a fila (Queue).
- heap.h: deverá ser usado para descrever a estrutura do Heap a ser implementada e a assinatura das funções que irão manipular o Heap.
- heap.c: deve ser usado para implementar as funções assinadas no arquivo heap.h
- ManagerProcess.c: deve ser usado para implementar todas as funções que serão usadas na função main para realizar a simulação do gerenciador de processo.

O que deve ser implementado.

No arquivo heap.h deve ser implementada a estrutura de dados que irá representar o Min-Heap. O grupo terá que optar por uma das três formas de implementar o Heap, isto é, a implementação por estrutura dinâmica, estrutura linear estática ou estrutura híbrida. Independente de qual abordagem se opte para implementar os heaps, **a implementação deve focar no conceito de que cada node no heap deve conter no mínimo dois campos**, um sendo a key do node, que deve ser um valor obtido fazendo pela diferença entre time_to_kill e



time_used (leia os arquivos myProcess.h e myProcess.c para entender o que são essas variáveis). O outro campo dos nodes do heap deve ser uma variável do tipo Process.

No arquivo heap.c devem ser implementadas as funções para manipular o Min-Heap, obviamente as assinaturas dessas funções serão consequência da escolha da implementação da estrutura do heap. Entretanto, três funções específicas devem ser implementadas:

- convert_queue_to_heap: deve converter uma queue de processos na estrutura de Heap descrita no arquivo heap.h.
- convert_heap_to_queue: deve converter um heap para uma estrutura de Queue, sendo tal estrutura descrita no arquivo myProcess.h, leia bem o arquivo caso necessário.
- update_key: função que irá atualizar a key de um node específico do heap.
- is_empty: função que verifica se o Heap está vazio ou não.


Quaisquer outras funções que se fizerem necessárias também podem (e devem) ser implementadas, em especial todas as operações sobre Heap que foram apresentadas em classe notavelmente serão necessárias.

Como vai funcionar o simulador de processos?

O simulador deve ser implementado, em um loop do tipo While que deve continuar em execução até que o Heap fique vazio. A cada interação deste loop, a função execute_Process especificada no arquivo myProcess.h e implementada no arquivo myProcess.c, deve ser executada n vezes (com n ser um valor aleatório definido a cada interação), onde n deve ser um valor menor ou igual ao número do nodes na heap.

Cada execução da função execute_Process deve ser feita com um processo diferente, ou seja, se o n for, por exemplo, igual a 4, significa-se que deve ser executada a função execute_Process para cada node da posição inicial até a posição 4 no heap.

Após todas as execuções da função execute_Process, deve então ser executada a função update_key para atualizar os keys no Heap. Em seguida, deve ser executada uma versão



modificada da função que remove a raiz do heap. Essa função modificada deve funcionar da seguinte forma, se a key na raiz tiver valor igual a 0 a raiz do heap é deve ser removida, por outro lado, no caso contrário, a remoção não deve ser realizada. Como a cada interação do loop, múltiplos nodes têm seu valor de key atualizado, é possível (mas não garantido) que múltiplos nodes fiquem com a key com valor zero, tenha isso em mente na sua implementação!

O que será avaliado?

A avaliação será dividida em duas partes, (1) a avaliação da qualidade de código, ou seja, a organização, clareza e corretude dos códigos e (2) a capacidade e criatividade do aluno, enquanto programador, no uso de uma linguagem de programação da sua escolha.