**Memory Management**

O código fonte foi realizado em apenas um único arquivo contendo tudo. Neste arquivo há uma class MemoryManagement sendo o namespace do código, contendo algumas variáveis globais, classes e funções (sendo uma delas a main).

A ideia é fazer com que processos fossem alocados na memória em uma lista, de base 2, no caso foi fixado expoente 6. Sendo o primeiro processo definido como “P1”. Cada elemento da lista representa o valor de tamanho 1 (um) na memória.

A lista, do tipo string, é atribuída com todos elementos internos como “0”, que será os espaços que estarão vazios na memória.

A class Process possui os métodos:

* announce(): Tem como objetivo de apresentar o seu nome e tamanho em tela.
* firstFit(): Tem como objetivo alocar o nome do processo em cada elemento da lista conforme o seu tamanho, priorizando o primeiro elemento vazio (“0”).
* nextFit(): Tem como objetivo alocar o nome do próximo em seguida do último inserido, e caso não foi possível, utiliza-se firstFit().
* bestFit(): Tem como objetivo em alocar o processo no memória onde o vão é o menor possível para o processo.
* worstFit(): Tem como objetivo em alocar o processo na memória onde o vão é o maior possível para o processo.

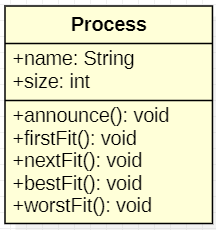


Figura 1: Classe Process.

Já em nas propriedades da class Process:

* Nome do processo, começando com P1 e em seguida P2, P3, P4...
* Tamanho do processo sendo um tamanho aleátorio.

Já no namespace, de maneira global, possuem as funções removeProcess() que remove aleatoriamente um processo. A função printMemory(), que renderiza a memória em tela. E a função main que instancia os objetos em um intervalo de 5 segundos.