



Relatório Escalonador

Implementação de um Escalonador

Sistemas Operacionais

*Mairon Schneider; Matheus Antonioli
maironcardoso;mgantonioli@inf.ufpel.edu.br*

Detalhes de implementação: Como foi feita a implementação?

Resposta: A implementação foi realizada utilizando a linguagem de programação C++, visto que, permite o uso de bibliotecas que já implementam estruturas complexas (Llist, Vector e Array), além disso, a linguagem C++ nos possibilita a usar a unidade Processos como um objeto que contem o slice de chegada, de duração, lançamento(que funciona como uma flag), o acumulador de slices(utilizando essa informação é possível trocar a prioridade do processo, caso necessário), o espaço que ocupa na memória, sua prioridade, e sua prioridade quando ele é lançado(inicial), o contador observado(observa em quanto tempo ele será executado), o marcador de ciclo(funciona como flag para saber se o processo já foi executado no ciclo).

Detalhes de métodos: Como foi feita a implementação dos métodos do objeto "Processo"?

Resposta: A implementação dos métodos é em grande parte métodos de gets e sets, porém, existe um método chamado de CPU, onde ele é responsável por diminuir um slice de um processo. Caso o acumulador de slices chega a 10, o processo tem sua prioridade reduzida sendo colocado em outra posição da Array de Llist (Lista de prioridades). Outro método necessário é o Stringsplit, que é responsável por separar os argumentos de entrada do algoritmo.

Detalhes do Algoritmo: Como foi feita a implementação da main.cpp?

Resposta: A implementação do main.cpp se dá primeiramente pela atribuição aos elementos do argv[], utilizando um atoi é possível receber as strings e atribuir no objeto processo. É instanciada no disco(memória onde é estanciada os processos), é utilizado um while para que o programa rode até o disco esvaziar. O for interno ao while, incrementa o tempo do for observado, sendo que os novos processos gerados são inseridos ao fim da llist. Há um for que testa se o processo pode ser adicionado na lista de prioridades, além disso, há um for que compara se a memória usada é maior que a memória requerida, descartando o processo, além disso, há um teste que determina se o processo pode ser adicionado a CPU.

Como executar: Como é feita a compilação e execução do programa?

G++ class.cpp main.cpp -o executavel

./executavel <numero de CPUS> <Espaço de memória> <arquivo.txt>

Testes: Utilizando um diferentes valores de cpu e mesma memória?

Resposta: Utilizando 256 de memória e variando o número de CPUS.

--Numero de CPUS: 1

--Quantidade de memory: 256

Chegada:7, Lancamento: 7, Duracao Projetada: 50, duracao observada: 51

Chegada:0, Lancamento: 0, Duracao Projetada: 10, duracao observada: 65

Chegada:5, Lancamento: 6, Duracao Projetada: 35, duracao observada: 101

Chegada:10, Lancamento: 70, Duracao Projetada: 20, duracao observada: 106

--Numero de CPUS: 2

--Quantidade de memory: 256

Chegada:0, Lancamento: 0, Duracao Projetada: 10, duracao observada: 11

Chegada:10, Lancamento: 10, Duracao Projetada: 20, duracao observada: 21

Chegada:7, Lancamento: 7, Duracao Projetada: 50, duracao observada: 51

Chegada:5, Lancamento: 57, Duracao Projetada: 35, duracao observada: 88

--Numero de CPUS: 3

--Quantidade de memory: 256

Chegada:0, Lancamento: 0, Duracao Projetada: 10, duracao observada: 11

Chegada:10, Lancamento: 10, Duracao Projetada: 20, duracao observada: 21

Chegada:7, Lancamento: 7, Duracao Projetada: 50, duracao observada: 51

Chegada:5, Lancamento: 57, Duracao Projetada: 35, duracao observada: 88

Alterando apenas o espaço de memória:

--Numero de CPUS: 1

--Quantidade de memory: 128

Chegada:7, Lancamento: 7, Duracao Projetada: 50, duracao observada: 51

Chegada:0, Lancamento: 0, Duracao Projetada: 10, duracao observada: 61

Chegada:10, Lancamento: 60, Duracao Projetada: 20, duracao observada: 71

--Numero de CPUS: 1

--Quantidade de memory: 256

Chegada:7, Lancamento: 7, Duracao Projetada: 50, duracao observada: 51

Chegada:0, Lancamento: 0, Duracao Projetada: 10, duracao observada: 65

Chegada:5, Lancamento: 6, Duracao Projetada: 35, duracao observada: 101

Chegada:10, Lancamento: 70, Duracao Projetada: 20, duracao observada: 106

--Numero de CPUS: 1

--Quantidade de memory: 256

Chegada:7, Lancamento: 7, Duracao Projetada: 50, duracao observada: 51

Chegada:0, Lancamento: 0, Duracao Projetada: 10, duracao observada: 65

Chegada:5, Lancamento: 6, Duracao Projetada: 35, duracao observada: 101

Chegada:10, Lancamento: 70, Duracao Projetada: 20, duracao observada: 106