UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ

INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS

FACULDADE DE ENGENHARIA DA COMPUTAÇÃO

RELATÓRIO REFERENTE À 2ª AVALIAÇÃO DE SISTEMAS OPERACIONAIS

MARAPÁ/PA – 2019

UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ

INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS

FACULDADE DE ENGENHARIA DA COMPUTAÇÃO

GABRIEL OLIVEIRA MACHADO

ATHOS LIMA ALCÂNTARA

JULIANA BATISTA DA SILVA

IAGO COSTA DAS FLORES

Trabalho apresentado como forma de obtenção de nota parcial na disciplina de sistemas operacionais, ministrada pelo Professor João Victor, no período 2019.4.

MARABÁ/PA – 2019

Sumário

[1 – INTRODUÇÃO 4](#_Toc23931193)

[2 – RESULTADOS E DISCUSSÕES 4](#_Toc23931195)

[3 – CONCLUSÃO 4](#_Toc23931196)

## 1 – INTRODUÇÃO

Para o cumprimento dos requisitos proposto onde o grupo dispõe o desenvolvimento simples para um escalonador circular também conhecido como Round-Robin, o mesmo se caracteriza com as fatais que utiliza de tempo que são atribuídas para os processos em partes iguais, e em ordem circular, esse processo pode ser chamado de processo cíclico.

De modo a trabalhar os termos discutidos foi implementado com ajuda da linguagem C ponteiros, filas e vetores, buscando categorizar e escalonar cada processo, já que sua forma geral de funcionamento é definida de buscando usar uma determinada quantidade de tempo que é definida pelo sistema operacional, que também determina o período de tempo entrada os sinais sem interrupções, esses processos são armazenados em uma fila circular.

## 2 – RESULTADOS E DISCUSSÕES

Tendo iniciado os devidos preparos para a compreensão do funcionamento teórico do escalonador, foram implementados os devidos includes de bibliotecas para que tudo pudesse ocorrer como o planejado. Para essa aplicação, é imprescindível a utilização da biblioteca QUANTUM, a mesma é utilizada visando deixar como responsabilidade do SO determinar o período de tempo sem interrupções.

Primordialmente temos a criação de uma fila denominada de “f”, está é a fila de processos, que inicialmente está vazia, sequencialmente temos a função que fica responsável por criar processos aleatoriamente, essa criação é feita seguindo a referência solicitado, quando a demanda é suprida, os processos criados são inseridos na fila citada. Vale lembrar que os processos têm características peculiares, sendo elas id, tipo, tempo de execução e estado, isso ocorre para cumprir que as características de um processo circular sejam visíveis e viáveis a implementação, como por exemplo, o estado do processo, que pode se encontrar em pronto ou bloqueado, ou ate mesmo para que o modo de execução saiba o tempo que esse processo vai ocupar. Isso tudo é gerado aleatoriamente buscando simular a realidade da aplicação em um sistema operacional.

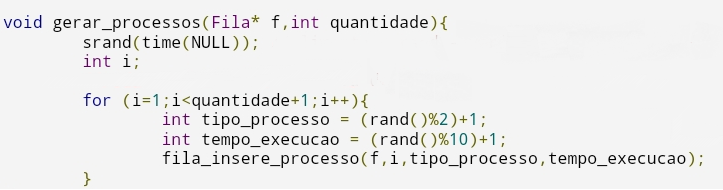


Figura 1: Função de gerar processos

Quando esse roteiro é cumprido, obtemos um ciclo fundamental para o funcionamento do escalonador já com os processos devidamente inseridos na fila com suas respectivas características, é possível entrar na próxima etapa onde os processos que já foram finalizados são retirados da fila e inseridos em outra, isso ocorre para manipular a fila anterior, supondo que ela vai estar vazia. Esses estados podem ser considerados como uma forma de remanejamento.

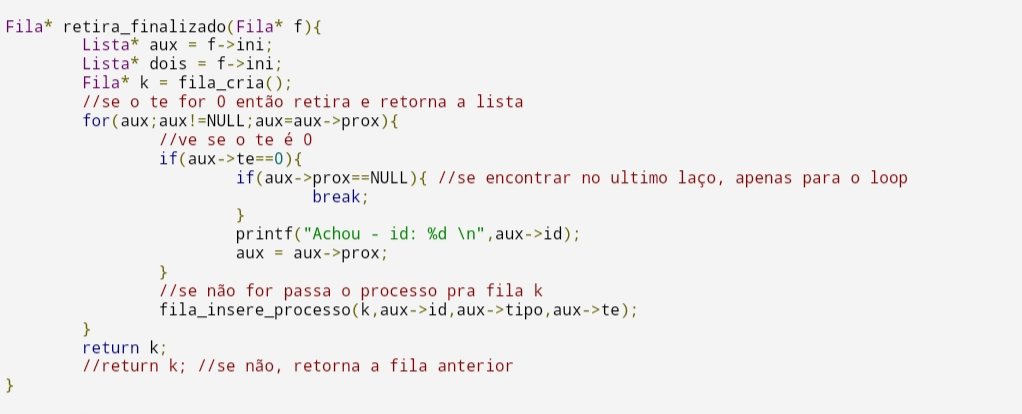


Figura 2: Retirar processo finalizado

Para e execução dos processos é possível lembrar que após o termino tempo que o mesmo vai passar desenvolvendo suas atividades é necessário subtrair o tempo atual do processo, retirando o mesmo e passando para o próximo processo.

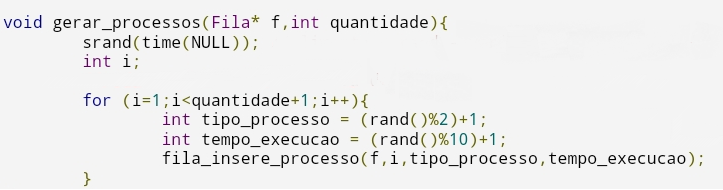


Figura 3: Gerar processo na fila de pronto

Por fim tudo isso existe para complementar a função principal, que é onde tudo acontece, já que existem necessidades a serem cumpridas, agora temos a possibilidade de criar processos, observá-los em execução, remanejar e até mesmo fazer a retirada deles quando terminam suas demandas.

Isso vai acontecer inicialmente com a alimentação das características individuais de cada um, para isso são chamadas as funções de criações de valores aleatórios para ajudar com esse requisito, os processos são exibidos, demonstrando suas características, seguindo as necessidades impostas pela variável aleatória, possibilitando sua retirada e remanejamento.

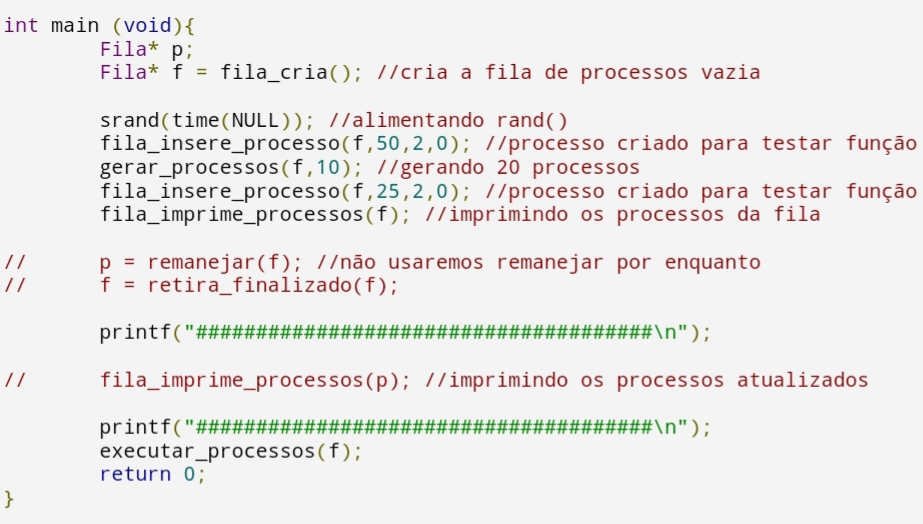


Figura 4: Função principal do código.

Função MAIN

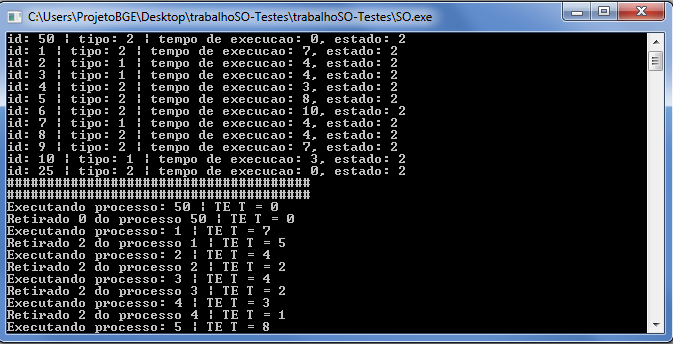


Figura 5: Execução do Processo

Na figura 5 demonstrou-se a execução do código do escalonador circular onde foram criados 12 processos automaticamente e foram dados valores aleatórios de tipo (entre I/O-Bound e CPU-Bound), tempo de execução (entre 1 a 10), e estado pronto para todos os processos.

Após isso o algoritmo começa a executar os processos na ordem que estão dentro da fila. Retirando a cada execução dois “Quantum” do tempo de execução para o processo executado com isso quando o processo chegar em zero ele é eliminado da fila. O programa continua essa rotina até finalizar todos os processos da fila.

## 3 – CONCLUSÃO

Foi desenvolvido um escalonador circular Round-Robin em linguagem c, que cria uma fila de processos e simula a execução deles no processador fazendo a alternância entre os estados de pronto, bloqueado e execução dos processos. Com isso, no algoritmo foi implementado cada processo como uma estrutura que recebe um id tipo, tempo de execução, estado do processo e um ponteiro que aponta para o próximo processo.