Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - IFAM
Campus Manaus Zona Leste
Disciplina: Sistemas Operacionais
Professor: Albert França Josuá Costa
Aluna: Brenda Cristina Dourado Moura

INTRODUÇÃO

Sistemas Operacionais (SO) são programas que agem na camada intermediária entre o computador e o hardware, realizando duas funções básicas. A primeira função é servir como máquina virtual, abstraindo assim detalhes de programação de hardware. Outra função bastante importante do SO é gerenciar processos (programas em execução). Um subtópico importante relacionado aos sistemas operacionais é relacionado aos processos e seus escalonadores, é importante entender que os escalonadores são os responsáveis por definir qual será o próximo processo que utilizará os recursos do processador, Existem quatro tipos básicos de escalonadores: O primeiro é o escalonador do tipo FIFO, como é sugerido por seu nome, a ordem de utilização do processador é definida pela ordem de chegada dos processos ao escalonador, este tipo possui como maior desvantagem o fato de que processos de menor importância e maior duração prejudiquem o funcionamento do processo, caso sejam adicionados primeiro ao processador. O próximo tipo é o escalonador por prioridade, nesse tipo o importante é a prioridade atribuída ao processo, processos com maior prioridade são executados primeiro. O terceiro é o Shortest Job First, nesse escalonador o processo possui um tempo e o escalonador é responsável por ordenar os processos a partir do menor tempo, ou seja, processos com o menor tempo atribuído será executado primeiro. O último é conhecido por Circular Round Robin, nesse tipo de escalonador o processo, além de possuir o tempo total de execução, também possui um atributo chamado de Quantum, esse atributo define o tempo-limite de utilização continua do processador, ao finalizar esse tempo o processo, caso ainda não tenha finalizado o tempo de execução, volta à fila de aptos. Para o bom funcionamento e implementação das filas de uso do processador, é importante compreender os estados que um processo pode assumir, sendo eles: criado, apto, executando, bloqueado e destruído. Este trabalho propõe construir uma simulação do funcionamento dos escalonadores do tipo FIFO, por Prioridade, Shortest Job First e Circular Round Robin. Esta abordagem de aprendizado prático visa estimular o aprendizado acerca do real funcionamento dos escalonadores.

ATIVIDADE DESENVOLVIDA

O objetivo principal do trabalho é criar os algoritmos de escalonamento dos tipos FIFO, por Prioridade, Shortest Job First e Circular Round Robin, simulando a execução dos processos por meio da escrita da string “Processo + identificador_do_processo” em um arquivo de texto a cada interação do programa, com o arquivo de texto não sendo sobrescrito.

Para a elaboração do trabalho foi criado uma estrutura híbrida, juntando conceitos de fila encadeada e árvore, dessa forma não foi necessário a criação de uma struct auxiliar com informações separadas sobre a lista encadeada. A estrutura híbrida denominada “no” é constituída por identificador, quantum, tempo, prioridade, estado, ponteiro para o pai, ponteiro para o primeiro filho, ponteiro para o próximo irmão, ponteiro para o próximo e para o nó anterior da lista encadeada.

Em relação ao código, ele é composto por quatro funções principais, sendo elas:

- tornarApto() – Criada para transformar em aptos os processos que estão na fila de criados.
- escreveDocumento() – Criada para acrescentar um texto (“Processo + identificador_do_processo”) em um arquivo de texto.
- executando() – Criada para executar a simulação de andamento de processos no ciclo do processador.
- main() – Função principal do programa, responsável por verificar informações de leitura de processos, verificação da opção do menu que foi escolhida pelo usuário e outras.

As funções anteriores são comuns à todos os escalonadores, com apenas pequenas diferenças. Como cada escalonador é único, é importante pensar nas particularidades de cada um, para isso, verificamos que os escalonadores e suas diferenças de implementação mais notáveis são:

- Tipo FIFO: para esse tipo de escalonador os processos utilizaram a parte referente à lista encadeada na estrutura híbrida, sendo os processos adicionados no modelo de uma fila.
- Tipo por Prioridade: o diferencial desse escalonador é a ordenação por prioridade no momento de adicionar na lista encadeada. Processos com a maior prioridade são adicionados mais para o início da lista.
- Tipo Shortest Job First: como o tipo de escalonador anterior, foi necessário que a lista fosse ordenada, nesse escalonador, os processos foram ordenados pelo tempo, ou seja, processo que possuíam o menor tempo total para execução eram adicionados mais para o início da lista.
- Tipo Circular Round Robin: nesse escalonador foi necessário utilizar o quantum para limitar o período que o escalonador poderia utilizar o processador, quando o tempo era finalizava e o processo ainda não havia concluído sua execusão, o processo era temporariamente “congelado” e voltada à fila de aptos.

RESULTADOS E CONCLUSÃO

Ao concluirmos a implementação dos algoritmos de escalonamento, pudemos verificar que o algoritmo FIFO possui maiores chances de possuir o pior desempenho e que isso ocorre quando processos com maior tempo de execução são adicionados primeiramente na fila de execução e que o mesmo pode acontecer quando os processos de maiores prioridades também possuem os maiores tempos de execução ao utilizarmos o escalonador do tipo por prioridade.

Em relação ao escalonador do tipo Shortest Job First não encontramos a situação anteriormente mencionada, já que esse escalonador é ordenado diretamente pelo tempo de execução, mas encontramos como principal ponto de atenção o fato de que processos importantes, mas com tempo de execução longo, podem demorar para que sejam executados, podendo atrapalhar todo o andamento de programas e/ou outros processos que possuem esse processo como pré-requisito.

O último escalonador, Circular Round Robin, aparentemente apresenta o melhor desempenho, pois o valor de quantum pode possibilitar que processos com menor prioridade e menor tempo de execução usem por mais tempo o processador e códigos com maior prioridade e maior tempo de execução possam usar o processador, mas por um período reduzido. Dessa forma, é mais provável que nenhum processo seja “esquecido” pelo escalonador, mas ainda assim tenham acesso ao ciclo do processador.

REFERÊNCIAS

https://www.seduc.ce.gov.br/wp-content/uploads/sites/37/2011/10/redes_de_computadores_sistemas_operacionais.pdf (Acesso em 15/11/2021, às 18h)

http://proedu.rnp.br/bitstream/handle/123456789/711/Sistemas_Operacionais_web.pdf?sequence=3&isAllowed=y (Acesso em 15/11/2021, às 19h)

https://labdegaragem.com/forum/topics/converter-char-para-int (Acesso em 16/11/2021, às 13h)

http://linguagemc.com.br/exibindo-data-e-hora-com-time-h/ (Acesso em 19/11/2021, às 14h)

https://www.inf.ufpr.br/wagner/so/processos+threads.2pp.pdf (Acesso em 20/11/2021, às 19h)

https://www.ti-enxame.com/pt/c/como-dividir-uma-string-em-2-strings-em-c/968225763/ (Acesso em 20/11/2021, às 18h)

https://www.ti-enxame.com/pt/c/como-remover-o-caractere-em-um-vazio-indice-de-uma-string-em-c/971615844/ (Acesso em 20/11/2021, às 18h)

http://wiki.icmc.usp.br/images/8/82/Manipulacao_arquivos.pdf (Acesso em 21/11/2021, às 19h)

http://www.facom.ufu.br/~gustavo/ED1/Apostila_Linguagem_C (Acesso em 24/11/2021, às 18h)

https://www.inf.pucrs.br/~pinho/LaproI/Structs/Structs.htm (Acesso em 24/11/2021, às 19h)

https://www.treinaweb.com.br/blog/ponteiros-em-c-uma-abordagem-basica-e-inicial (Acesso em 24/11/2021, às 20h)

https://www.inf.pucrs.br/~pinho/LaproI/Structs/Structs.htm (Acesso em 25/11/2021, às 18h)

https://www.gsigma.ufsc.br/~popov/aulas/so1/cap8so.html (Acesso em 25/11/2021, às 20h)