

RELATÓRIO PROJETO NÚCLEO SISTEMAS OPERACIONAIS 2

Antônio Eugênio RA:161021336
Thiago Ho art Vieira RA:161026524
Lucas Vinhas RA:161023584

Junho/2019

RESUMO

Este relatório apresenta as atividades realizadas na construção do núcleo multitarefas em uma máquina virtual na disciplina de Sistemas Operacionais 2 ministrada pelo Profº Drº Antônio Carlos Sementille no primeiro semestre de 2019. Este projeto conta com o núcleo construído em 4 fases sendo: A primeira o núcleo básico, a segunda o núcleo básico com a adição das funções de semáforo, a terceira sendo o núcleo básico com adição de troca de mensagens, e a quarta o núcleo básico com prioridades. Todavia os núcleos são simulados em processador 8086 utilizando o Turbo C num nível básico apoiados em co-rotinas, cada uma delas com suas respectivas particularidades.

1 Núcleo 1

O 1º núcleo conta com o escalonamento de processos feito através do algoritmo Round Robin e as funções básicas de um núcleo. A principal dificuldade enfrentada em seu processo de implementação foi o entendimento do funcionamento e das limitações do Turbo C, tal como as implementações das funções básicas para o funcionamento do núcleo. Segue a lista das funções e variáveis implementadas e uma descrição de seu funcionamento:

- APONTA_REG_CRIT - Aponta para a região crítica do DOS.
- DESCRITOR_PROC - Estrutura que define as informações de um processo, que contém nome, estado, contexto e um ponteiro para o próximo descritor.
- prim - Variável global usada como lista circular, que aponta para o primeiro da lista.
- procura_proximo_ativo() - Função que utiliza o Round Robin, verificando os estados dos processos para pegar o próximo ativo.
- volta_dos() - Altera os valores de interrupção e volta do DOS.
- cria_processo() - Cria um processo e recebe estes parâmetros de inicialização e adiciona o processo para a lista circular.
- dispara_sistema() - Inicia o sistema
- terminar_processo() - Finaliza o processo e volta ao DOS caso não tenha processo ativo.
- escalador() - Escala processos, sendo responsável pelo controle da região crítica do DOS e escalonamento dos processos ativos.

Para o teste implementamos 3 processos que são loops, onde o primeiro e segundo processo tem a mesma quantidade de interações, e o terceiro tem mil a mais, de forma que eles sejam escalonados de acordo com o algoritmo.

2 Núcleo 2

O 2º núcleo apresenta a implementação de semáforos, utilizado para a exclusão mútua. Utiliza-se semáforos em processos que se encontram na mesma máquina e que tem acesso a alguma região ou variável compartilhada. Utilizam duas primitivas, up e down. Além disso, foi adicionado um campo de fila de semáforo, que aponta para o próximo processo bloqueado na fila de semáforo.

As funções adicionadas em relação ao primeiro núcleo foram:

- inicia_semaforo() - Inicia o semáforo com o valor desejado.
- up() - Desbloqueia o primeiro processo da fila caso existam processos bloqueados, avança na fila de processos, e caso a fila esteja vazia, libera a variável do semáforo para acesso.
- down() - Verifica se o processo tem acesso permitido, se não for permitido, o bloqueia e coloca na lista de bloqueados pelo semáforo.

Para testar o funcionamento do suporte de semáforo, foi implementado o Problema do Produtor/Consumidor atuando sobre um Buffer. O teste funciona da seguinte maneira: O produtor enche o buffer, após o buffer estar cheio, o consumidor começa a consumir o buffer. Após o consumidor ser finalizado, é gerado um arquivo teste2.txt com todas as interações.

3 Núcleo 3

O 3º núcleo apresenta as trocas de mensagens, ou seja, um mecanismo de comunicação que exige do Sistema Operacional a sincronização de seus processos e, também, a comunicação entre eles. Sendo eficiente para comunicação entre computadores diferentes e comunicação em um mesmo computador, já que garante a exclusão mútua. Para que isso ocorra, foi alterada a estrutura do descritor de processos para poder alocar dados referentes às mensagens, campos foram adicionados, como tamanho da fila permitida, mensagem e quantidade atual de mensagens, também foi alterada a função de criação de processos, para o processo poder ser inicializado com os dados das mensagens. Na implementação, foi adicionada duas operações bloqueantes fundamentais, são elas: envia e recebe. As funções tem lógicas parecidas, buscando uma mensagem enviada ou recebida e mudando o estado do processo. Segue as funções e variáveis criadas para o funcionamento do 3º núcleo:

- envia() - Responsável pelo envio de uma mensagem de um processo à outro. O processo que envia a mensagem é bloqueado e o que recebe a mensagem, desbloqueado, se necessário, e a atualização na fila de mensagens é realizada.
- recebe() - Responsável pelo recebimento de uma mensagem de um emissor qualquer.

Para testar o funcionamento da troca de mensagens, foi implementado dois processos trocando mensagens entre si. É gerado um arquivo teste3.txt com todas as interações.

4 Núcleo 4

Nesta etapa, aproveitamos a estrutura definida no núcleo 1, pois não é necessário as funções criadas pelo núcleo 2, e adicionamos campos de prioridade, tanto na estrutura do descritor de processos, bem como na função que cria os processos. Também no escalador, mantemos o processo em execução por uma quantidade de tempo (quantum) proporcional à prioridade atribuída. Dessa forma, processos com maior prioridade tem mais tempo de execução. Então no teste 4, criamos três processos com a mesma instrução e com prioridades diferentes e notamos que os processos com maior prioridade terminam mais rápido. No caso em questão, o processo 3 tem prioridade 5; o processo 2 tem prioridade 2; o processo 1, prioridade 1.

5 Conclusão

Nesse trabalho conseguimos implementar um núcleo básico de um Sistema Operacional, emulado no processador 8086 utilizando o DosBox e o TurboC. Foram implementados quatro núcleos, sendo eles: com Processos, Semáforo, Troca de mensagens entre processos

e processos com prioridades. Com esse trabalho conseguimos obter uma ideia boa de como um Sistema Operacional funciona de uma forma simplificada, organizando os processos e controlando seus recursos.