PUCRS - Escola Politécnica
Disciplina: Sistemas Operacionais - 2021/2 - Trabalho Prático - Fase 5
Prof. Fernando Luís Dotti

Nesta fase do trabalho introduzimos o gerente de processos (GP) incluindo o escalonamento de processos.

1. Gerente de Processos

O GP é um módulo do SO e é responsável por

- Criar um processo, dado um programa passado como parâmetro.

boolean criaProcesso(programa)
verifica tamanho do programa
pede memória
se nao tem memória, retorna negativo
Cria PCB
Seta tabela de páginas no pcb
Carrega o programa nos frames alocados
Seta demais parâmetros do PCB (id, pc=0, etc)
Coloca processo na fila de prontos
Retorna true

- Desloca um processo

deslocaProcesso (id)
desloca memória do processo com id
retira de qualquer fila que esteja
desloca pcb

2. Escalonamento

Adotaremos escalonamento baseado em fatia de tempo, devido à simplicidade. Isto significa que processos executarão por um período na CPU. Decorrido este período o mesmo deixa a CPU e outro processo deve executar. Esta função de escalonamento é crucial em sistemas multiprogramados. Para tal se faz necessária:

1. Uma forma de interromper a execução de um processo na CPU, decorrido um tempo;
2. Ter uma rotina de salvamento do estado atual do processo executando, que perde a CPU. Este estado deve conter todas informações importantes da CPU, necessárias à continuação do processo quando for escalonado novamente. Este estado deve ser salvo no PCB.
3. Criar uma fila de processos prontos;
4. Ter uma rotina de escolha de um processo pronto, para executar;
5. Ter uma rotina que restaura o estado do processo escolhido na CPU e retoma sua execução.

Abaixo alguns comentários específicos.

2.1. Retomada de controle com a passagem do tempo

Em um sistema real existe um relógio independente no HW. Este relógio interrompe periodicamente a CPU. A CPU desvia para a rotina de tratamento do relógio. Entre outras funções, esta rotina avalia se o processo que está rodando deve ser trocado. No nosso sistema podemos implementar o relógio como um contador de instruções executadas na CPU. A cada C instruções executadas na CPU, esta interrupção acontece (por exemplo C pode ser 5).

2.2. Funcionamento geral

Resolvida a forma de criar um sinal de relógio (em 1 acima). Deve-se criar uma nova interrupção no sistema, e com ela uma nova rotina de tratamento para a mesma. Esta é a interrupção ativada pelo relógio a cada C instruções executadas. A sinalização e desvio para interrupção acontece exatamente como nas anteriores.

A rotina de tratamento deve salvar o estado da CPU no PCB do processo executando. Ou seja, a cada momento deve-se saber qual processo está *running* e quais estão *ready.* O SO pode ter um ponteiro para o PCB do processo *running* e uma fila de processos *ready.* Quando um processo sai da CPU, ele sai de running e vai para o fim da fila ready.

A escolha de outro processo a executar pode ser o primeiro processo da fila ready. Toma-se o PCB do processo escolhido, que fica running, e restaura-se o estado de execução deste PCB na CPU. Ao final, a CPU retoma a execução, agora no novo processo. Este processo executará até que uma nova interrupção do relógio seja sinalizada, e o comportamento descrito aqui se repete.

3**. Testes**

Carregue vários processos em memória. Estes processos estão todos parados, na fila ready.
Então dispare o processo de escalonamento, que escolhe um processo e coloca na CPU.
A partir disso, o ciclo exposto em 2 deverá acontecer até que todos processos acabem.