

Exercício Programa II

MAC0422 _ sistemas Operacionais

Prof. Alan M. Durham

Entrega:02/05/2023, 07:59hs

O sistema de escalonamento de processos do MINIX é multinível: prioridades iniciais, que podem ser alteradas durante a execução, são atribuídas aos processos. Processos do usuário são iniciados com a mesma prioridade, que é menor do que a prioridade de qualquer processo do sistema. O escalonador gerencia 16 filas de processos de diferentes prioridades utilizando as estruturas *rdy_head* e *rdy_tail*. **Você pode ler mais detalhes sobre a implementação do escalonamento do Minix na seção 2.6.10 do livro.**

Neste EP, você deve modificar o sistema operacional MINIX, e fazer duas tarefas

1. Fazer com que os processos das filas gerenciadas pelo escalonador sejam mostrados na tela quando a tecla F4 for pressionada. Os atributos dos processos, que devem estar ordenados pela prioridade de execução, a serem impressos são:

- prioridade de execução;
- *process id*;
- tempo de cpu;
- tempo de sistema;
- endereço do ponteiro da pilha.

2. Acrescentar uma nova chamada de sistema ao Minix:

- *lockpriority(pid, priority)*, que permita a um processo mudar a prioridade de seus processos-filhos. Atenção, a chamada deve verificar se o pid é de um processo filho e se a prioridade é prioridade de usuário. A rotina volta a nova prioridade quando a mudança é bem sucedida, -1 se a prioridade é inválida e -2 se o processo indicado por *pid* não é processo filho. A partir desta chamada a prioridade do filho deve ficar INALTERADA. Esta última característica deve ajudar você a testar o programa.

Cuidado: para tornar uma prioridade não mutável pelo escalonador você deve indicar isso na tabela de processos, em algum campo. Se precisar adicionar campos na tabela do kernel, deve tomar MUITO cuidado para verificar se as constantes de tamanho do código do Kernel não precisam ser mudados

DICA: uma nova chamada de sistema envolverá uma nova rotina na biblioteca de sistemas que transformará a chamada em uma mensagem para o Process Manager. Este deve tratar a nova mensagem, mudando as prioridades com a ajuda do SystemTask.

3. Mudar o algoritmo de escalonamento utilizado pelo Minix para as filas de prioridade 7 a 14. Nesta nova versão você vai fazer um round-robin modificado multi-nível. Neste novo esquema, cada fila funcionará de maneira "round-robin", porém o algoritmo para escalonar processos de menor prioridade será diferente. Toda vez que todos os processos da camada *N* são escalonados, um processo da camada *N+1* é escalonado, e assim por diante. Para esclarecer vejamos um exemplo: Suponha que tenhamos apenas três filas, com os seguintes processos, em ordem decrescente de prioridade:

1. P1, P2, P3
2. P10, P15, P16
3. P18, P19

Neste caso a ordem de escalonamento seria

P1,P2,P3,P10,P1,P2,P3,P15, P16, P18,P1,P2,P3,P10,P1,P2,P3,P15,P1,P2,P3,P16,P19

Note que automaticamente os processo com filas com menor prioridade são executados menos vezes, mas ao contrário do multi-level feedback queues visto em classe, todas as filas de prioridade tem garantia de execução independente do comportamento dos programas.

Para facilitar os testes, mude também a estratégia de desbloqueio dos processos. No Minix atualmente, quando um processo que não usou todo o seu quantum de processamento é desbloqueado, ele entra no início da sua fila de prioridade. Em sua versão faça ele sempre ir para o final da fila e reinicialize seu quantum. Com isso fica mais fácil prever o comportamento de programas. Processos que tem sua prioridade fixada não devem mudar de prioridade, ou seja continuam com a mesma, independente do que fizerem.

NOTA1:Pense como o ep acima pode ser testado. Note que a chamada de sistema para fixar a prioridade fará que apenas comandos "print" podem fazer com que você consiga testar o funcionamento de seu escalonamento. Podemos discutir como el classe. Você pode tentar reproduzir a fila que mostramos no exemplo....

NOTA2: pense bem na sua solução. Você vai precisar saber, em cada fila, quais processos já foram escalonados uma vez, para poder controlar quando irá escalonar a fila de baixo. Talvez acrescentar um bit no descritor.

Detalhes Administrativos

1. O trabalho deve ser executado em duplas. Como regra, cada EP gerará uma nova imagem do sistema, mas as imagens serão sempre construídas em cima da imagem anterior, assim recomendamos que o grupo seja mantido no semestre. Mudanças devem comunicadas ao professor
2. Trabalhos atrasados terão uma penalidade de 20%. Só serão aceitas entregas até dois dias após prazo regulamentar.
3. No dia 004/05/2023 será realizada uma provinha sobre o EP, com nota entre 0,0 e 1,0. Sua nota final do ep será multiplicada pela nota da provinha**Entrega: você deve produzir uma imagem e um documento PDF.**
 - a. A imagem deve conter os novos arquivos fontes adicionados à biblioteca do sistema para implementar a nova chamada e o sistema com as modificações pedidas.
 - b. O documento pdf deve conter um relatório sucinto do que foi feito, incluindo localização dos arquivos novos e dos arquivos modificados.
 - c. Nos arquivos modificados o código novo deve estar bem ressaltado com linhas de comentário de '#' antes de depois das linhas modificadas, Ex:
/*#####*/
<codigo modificado>
/*#####*/
4. **IMPORTANTE:** a nota do seu ep será multiplicada pela nota obtida na provinha de ep que será realizada na data seguinte à entrega. As provinhas tem valor 1,0. Desta maneira, se você tirar 8,5 no EP e 0,5 na provinha, sua nota de EP será 4,25. As provinhas conterão questões que vão refletir seu conhecimento do problema, não a qualidade de sua solução.