

Primeiro Trabalho de INF 1019– 2013.1

Data de entrega: 10 de maio de 2013

Objetivo:

Neste trabalho voce deve implementar semáforos em MINIX. Para tal, voce deve adicionar as seguintes quatro novas chamadas de sistema:

- `int semDown (int SemaphoreID)`
- `int semUp (int SemaphoreID)`
- `int semInit (int SemaphoreID, int Value)`
- `int semStatus (int SemaphoreID, int* Value, int* NimBlocked)`

Essas chamadas devem ser tratadas no servidor **ProcessManager** (PM), ou seja, é lá que o estado dos semáforos deve ser mantido, bem como as filas de processos bloqueados para cada semáforo. O PM deve ser capaz de gerenciar até 10 semáforos, (**SemaphoreID** $\in [0,9]$), e cada um deve ser inicializado com um valor inicial através da chamada **semInit()**. As chamadas **semDown()** e **semUp()** são as operações tradicionais de consulta/bloqueio em um semáforo e de incremento/liberação de um semáforo, respectivamente. Por fim, **semStatus()** deve retornar o valor atual (Valor ≥ 0) do semáforo passado como primeiro argumento, e possivelmente o número de processos bloqueados no mesmo. Além disso, implemente também o tratamento de erro para as chamadas, isto é, verifique se **SemaphoreID** é um valor entre 0 e 9. Se não for o caso, a system call deve retornar -1 e setar a variável **errno** com a constante **EINVAL**.

Note que todo o gerenciamento de semáforos deve ser implementado através de envio de mensagens entre processos, ou seja, usando as funções **mini_send**, **mini_rec**, e **mini_notify** (todas em **kernel/proc.c**), implementadas pelo núcleo.

Criação de novas chamadas de sistema:

A criação de uma nova chamada de sistema, essencialmente consiste dos seguintes passos:

- escolha de um novo número, ainda não usado, para a chamada (em */usr/src/include/minix/callnr.h*), defina o protótipo de ponteiro de função (em *usr/src/servers/pm/proto.h*) e adicione a entrada correspondente na tabela de chamadas sendo tratadas pelo PM (*/usr/src/servers/pm/table.c*)
- a implementação do(s) procedimento(s) que faz(em) o tratamento da chamada e a especificação de seus protótipos em arquivos (*/usr/src/servers/pm/tratador.c* e */usr/src/include/tratador.h*, respectivamente) para permitir a sua inclusão no novo executável de PM. Note que a depender do número e do tipo dos parâmetros a serem passados para a chamada, voce deve escolher um dos 6 tipos de mensagem disponíveis (**mess_1** a **mess_8**, em *include/minix/ipc.h*);
- Criação dos procedimentos da biblioteca para chamda de sistema, por exemplo, **_semDown.c**, a ser colocada no diretório */usr/src/lib/posix/*. Este procedimento deve então preencher os campos da mensagem escolhida e fazer a chamada ao sistema propriamente dita, por exemplo, **_syscall (PM, 58, &m)**.
- Após compilar os procedimentos da biblioteca, acrescente o código objeto na biblioteca run-time de C */usr/lib/i386/lib.c*

Mais infromações e um passo-a-passo de como criar uma chamada de sistema no Minix 3 voce encontra na seguinte URL: www.inf.puc-rio.br/endler/courses/inf1019/minix/

Testes:

Juntamente com o seu Minix modificado, voce deve entregar um programa tese.c, que faz um fork & exec e que faz chamadas a SemInit, semDown, semUp, mostrando como os processos pai e filho podem se sincronizar usando semáforos, e que imprime o resultado de semStatus, em um arquivo (ou na saída padrão), para várias situações. Prepare testes para semaforos com diferentes valores iniciais.

Observações:

- Os trabalhos devem ser feitos em grupos de dois. Indique claramente os integrantes do grupo no cabeçalho dos arquivos alterados.
- Os grupos poderão ser chamados para apresentações orais/ demonstrações dos trabalhos entregues.

- Marque, no fonte dos arquivos alterados, os locais onde você fez alguma modificação (inclusão, remoção, ou modificação de comandos) com comentários iniciados com a palavra chave ALT
- Os trabalhos deverão ser entregues em forma digital (em CD-ROM) incluindo os fontes dos arquivos alterados (somente), os arquivos de teste e a nova imagem gerada do sistema.
- Acrescente também um arquivo ASCII contendo um pequeno relatório explicando, como o que está funcionando e, principalmente, o que não está funcionando.