

MAC0422 - Sistemas Operacionais - 2018

Relatório do EP 2

Danilo Escudero - 10853911
Renan Tiago - 9793606

ATIVIDADE 1

Para alterar a tecla f4 e imprimir as informações solicitadas, alteramos o arquivo `/usr/src/servers/is/dmp.c`, onde é mapeado o nome de uma função à tecla, e o arquivo `/usr/src/servers/is/dmp_kernel.c`, onde declaramos a função propriamente dita, que varre as estruturas `mproc` e `proc`, às ordena de acordo com a prioridade de cada processo, e imprime as informações necessárias.

ATIVIDADE 2

Para criar a chamada, modificamos a quantidade de chamadas (NCALLS) em `/usr/src/include/minix/callnr.h` e definimos a chamada `CHPRIORITY` com o número 31.

Adicionar o nome da chamada (com o prefixo `do_`) na lista de funções do arquivo `/usr/src/servers/pm/table.c`: `do_chpriority`.

Escrevemos a função `do_chpriority` da chamada no diretório `/usr/src/servers/pm/` em um novo arquivo, com o nome `chpriority.c`. Após isso, incluímos o nome do arquivo no `Makefile`.

Definimos um protótipo para a função em `/servers/pm/proto.h`.

Criamos o arquivo `_chpriority` em `/usr/src/lib/posix` com a função da chamada.

Criamos o header `/usr/src/include/chpriority.h`, com a função da chamada.

Criamos um arquivo com o nome `sys_chpriority.c` em `/usr/src/lib/syslib` e acrescentamos no `Makefile`.

No arquivo `/usr/src/include/minix/com.h` acrescentamos um `define` para a chamada `chpriority` e aumentamos a quantidade de chamadas de 31 para 32.

Criamos o arquivo `do_chpriority.c` em `usr/src/kernel/system` e o acrescentamos no `Makefile`.

Fizemos um `define` em `usr/src/kernel/config.h`.

Criamos um mapeamento entre `sys_chpriority` e `do_chpriority` no diretório `/usr/src/kernel/system.c`.

Criamos um PROTOTYPE em `usr/src/kernel/system.h`.

Criamos um PROTOTYPE em `usr/src/include/minix/syslib.h`.

Criamos um assembly que aponta para a chamada em `/usr/src/lib/syscall` com o nome `chpriority.s`.

Após isso, recompilamos o kernel e todas as bibliotecas do sistema com ***make world*** em `/usr/src` e ***make hdboot*** em `/usr/src/tools`.

Após todos esses passos, a system call e a kernell call estão prontas e se comunicam.

Por último, criamos um programa que usasse a chamada de sistema.