

Relatório - EP Fase 4

Laboratório de Programação 2

21 de setembro de 2014

1 Integrantes

- Victor Sanches Portella - N° USP: 7991152
- Vinícius Vendramini - N° USP: XXXXXXXX

2 Parte 1

!!!!

3 Parte 2

Para fazer a parte 2, criamos uma System Call chamada (`setpriority_ep(pid, pri)`), onde **pid** e `textbfpri` são inteiros representando respectivamente o pid do processo alvo e a nova prioridade desse processo.

Essa System Call cria uma mensagem, que é enviada para o Process Manager, que chama a função **do_setpriority_ep()**. Lá verificamos se o processo chamador é pai do processo alvo, além de testar se o PID passado é de fato válido. Caso passe nos dois testes, chamamos a Kernel Call **sys_nice**, que enviará uma mensagem para o System Task. Lá será verificado se a prioridade passada é válida, !!!!!além de alguns outros testes de segurança!!!!!!.

Caso não haja nenhum problema, a prioridade é mudada na table de processos do Kernel, e o processo é mudado de fila de prioridade.

É importante notar que a prioridade **pri** passada é um número entre -20 e 20, não representando a real fila para a qual o processo será re-alocado. Quanto mais alto for **pri**, maior será a prioridade do processo. Dado esse

número, o **sys_nice** faz uma conta para ter uma equivalência com relação ao número da fila que o processo deveria ir. Aqui fizemos uma tabela mostrando e qual fila o processo será realocado para cada valor de **pri**.
!!!!Falta tabela!!!!

4 Teste