SISTEMAS OPERACIONAIS – 2013/2 – Prof. José Gonçalves

Gabarito da Prova 3

Questão 1

Se execl() tem sucesso o programa imprime:

- Child runs
- Got a dog
- Parent done

Se execl() falha, será impresso:

- Child runs
- Child done
- Parent done

Questão 2

Para i=0 serão criados 5 processos, sendo que "Prova Final" será impresso 4 vezes.
Para i=1 serão criados mais 19 processos e "Prova Final" será impresso 18 vezes.
"Prova Final" será impresso 22 vezes no total.

<fazer a figura>

Questão 3

O programa ajuda a descobrir qual é o tamanho real do buffer usado na implementação de pipes pelo sistema operacional. O POSIX sugere 512 bytes mas sistema Unix frequentemente apresentam capacidade muitos superiores para esta área. Por exemplo, testando numa máquina com Ubuntu 10.04.4 (kernel 2.6.32-46-generic), este limite foi de 65536 bytes.

No programa em questão, após criar o pipe e instalar um novo tratador de sinal para SIGALRM, ele entra em um loop onde grava um caractere "x" no *pipe* a cada rodada. Caso o número de caracteres gravados seja múltiplo de 1024 o programa imprime este valor, por exemplo, "1024 chars in pipe", "2048 chars in pipe", etc. Quando o buffer enche, o processo é bloqueado ao tentar gravar no *pipe*. Nesse momento, a variável *count* contém o número total de caracteres gravados - e, portanto, o tamanho do *pipe*. Observe que imediamente antes de ficar bloqueado no *write*, o processo executou *alarm(20)*. Assim, decorrido este tempo, o tratador de sinal *alarm_action()* será executado, imprimindo "write blocked after xxxx chars", onde xxxx define o tamanho real do *pipe*.

Questão 4

V
F – suspende a execução da *thread* que invocou o comando.
V
V
F – não existe este comando na biblioteca *phtreads*.

Questão 5

Primeiramente, o programa inicializa uma nova máscara de sinais bloqueados (“newmask”) e adiciona SIGINT a esta máscara. O programa altera então a máscara de sinais para “newmask”, guardando a máscara antiga em “oldmask”. A partir deste momento, o programa passa a não reconhece mais o CRTL-C (comentário). Finalmente, o programa retorna com a máscara antiga (“oldmask”), sem guardar o conteúdo de anterior da máscara definida em “newmask”.

Questão 6

V
F – o arquivo não é removido quando o processo que criou o pipe é finalizado.
V

Questão 7

Cada página tem tamanho 200, suportando o armazenamento de 200 elementos da matriz. No caso do código de inicialização apresentado, cada página armazenará duas linhas da matriz. A cada duas iterações do comando “for” é necessário carregar uma nova página (é gerado um *page fault*) . Esse comportamento do código de inicialização resulta em um total de 5000 *page faults*.

Questão 8

a) Cálculo do enderço físico
Processo A
Página = Endereço virtual/Tamanho da página
Página = 12K/4K = 3
Deslocamento = 0
Moldura = 4 (analisando a tabela de páginas de A)
Endereço físico = número da moldura x tamanho + deslocamento = 4 x 4K + 0 = 16K.

Processo B
Analisando a tabela de páginas do processo B, verifica-se que a página 3 não está mapeada na memória. Haverá a ocorrência de um *page fault*.
<não precisava detalhar como o sistema resolve a situação>

b) Tabela de Páginas de dois níveis

<fazer figura>

Observar que precisaremos de uma tabela de primeiro nível e apenas três tabelas de segundo nível. No primeiro nível, a terceira entrada, referente aos endereços de 32-48K, estará vazia. No segundo nível, apenas a primeira tabela (de endereços de 0-16k) usará todas as entradas . A terceira tabela (32-48K) não será necessária pois o processo não referencia esse espaço de endereços virtual.