Pontifícia Universidade Católica do Rio de Janeiro

Lab 2 - Memória Compartilhada

Sistemas Operacionais

Alunos	Tharik Lourenço
	Suemy Inagaki
Professor	Luiz Fernando Seibel

Rio de Janeiro, 07 de abril de 2021

Conteúdo

1 Problema 1

1) Faça um programa paralelo para localizar uma chave em um vetor.

- Crie um vetor na memória compartilhada contendo 4K números inteiros gerados aleatoriamente, pertencentes ao intervalo [0,999] e divida o vetor pelo número de processos utilizados na busca (usar N processos, N¿=4). Cada processo deve procurar o dado na sua área de memória e informar as posições onde o dado foi localizado.

1.1 Código Fonte

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <unistd.h>
#include <sys/wait.h>

#define TAM_TOTAL 4000
#define NUMPROC 4
#define numP 80

int main( void ){

    int segmento,*p;
    segmento=shmget(IPC_PRIVATE, (TAM_TOTAL*sizeof(int))
    ,IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR);
    if(segmento<0){
        printf("Shmget erro!\n");
        exit(1);
    }
    p = (int*)shmat(segmento,0,0);
    for(int t=0;t<TAM_TOTAL;t++){
        if (t%100 ==0)
            p[t] = 80;
        else
            p[t]= rand()%1000;
    }
```

```
    for (int i = 0; i < NUMPROC; ++i)
    {
        if (fork() == 0){
            printf("começo do processo filho %d\n",getpid());

            /*separa igualmente a qtd de elementos para cada processo*/
            int start = 0;
            for(int j=0;j<i;++j){
                start += (TAM_TOTAL+1-start) /(NUMPROC - j);
            }
            int tam = (TAM_TOTAL+1-start) /(NUMPROC - i);
            int end = start + tam;

            for(int n = start ;n < end;n++){
                printf("processo %d (pai:%d)- pesquisando[%d] em %d : %d\n" ,
                getpid(), getppid(),numP, n,p[n] );
                if (numP == p[n]){
                    printf("processo %d (pai:%d)- achou o numero %d no
                    espaço %d\n" ,getpid(), getppid(),numP ,n);
                }
            }

            exit(EXIT_SUCCESS);
        }
    }
    for (int i = 0; i < 3; ++i)
        wait(NULL);
    printf("processo pai vai terminar !!!\n");
    exit(0);
}
```

1.2 Resultado da Execução

1.2.1 Comentários

- Definimos que vamos procurar o número 80 (#define numP 80) e forçamos que aparecesse em todas as posições multiplas de 100.
- O funcionamento do programa pôde ser verificado, pois como podemos ver no resultado abaixo, os processos filhos encontraram os números 80 que escolhemos, além dos eventuais 80s gerados aleatoriamente.

- O processo 1 busca nas posições de 0 a 999, o processo 2 busca de 1000 a 1999, o processo 3 busca de 2000 a 2999 e o processo 4 busca de 3000 a 3999

1.2.2 Resultado

```
$ gcc lab2.c -o lab2
$ ./lab2

come o do processo filho 78
processo 78 (pai:76)- pesquisando[80] em 1000 : 80
processo 78 (pai:76)- achou o numero 80 no espa o 1000
...
processo 78 (pacome o do processo filho 77
processo 77 (pai:76)- pesquisando[80] em 0 : 80
processo 77 (pai:76)- achou o numero 80 no espa o 0
...
processo 77 (pai:76)- pesquisando[80come o do processo filho 80
processo 80 (pai:76)- pesquisando[80] em 3000 : 80
processo 80 (pai:76)- achou o numero 80 no espa o 3000
...
processo 80 (pai:76come o do processo filho 79
processo 79 (pai:76)- pesquisando[80] em 2000 : 80
processo 79 (pai:76)- achou o numero 80 no espa o 2000
processo 79 (pai:76)- pesquisando[80] em 2001 : 916
...
processo 79 (pai:76)- pesquisando[80] em 1077 : 170
processo 78 (pai:76)- pesquisando[80] em 1078 : 956
...
processo 78 (pai:76)- achou o numero 80 no espa o 1100
...
processo 78 (pai:76)- pesquisando[80] em 1154 : 192
proces] em 80 : 750
processo 77 (pai:76)- pesquisando[80] em 81 : 87
...
processo 77 (pai:76)- achou o numero 80 no espa o 100
...
processo 77 (pai:76)- pesq)- pesquisando[80] em 3077 : 205
processo 80 (pai:76)- pesquisando[80] em 3078 : 21
processo 80 (pai:76)- pesquisando[80] em 3079 : 835
...
processo 80 (pai:76)- achou o numero 80 no espa o 3100
...
processo 80 i:76)- pesquisando[80] em 2077 : 974
processo 79 (pai:76)- pesquisando[80] em 2078 : 401
...
processo 79 (pai:76)- achou o numero 80 no espa o 2100
...
processo 79 (pai:76)- pesquisando[80] em 2154 : 249
processo 78 (pai:76)- pesquisando[80] em 1155 : 729
...
...
...
processo 80 (pai:76)- pesquisando[80] em 4000 : 80
processo 80 (pai:76)- achou o numero 80 no espa o 4000
processo pai vai terminar !!!
```

1.3 Perguntas

1. **Houve paralelismo? Como você justifica a sua resposta?** Sim! Podemos ver que os processos filhos foram executados simultaneamente, alternando entre um e outro, o que indica o paralelismo!