Sistemas Operacionais
TP07 - Processos com Memória Compartilhada
Marcelo Makoto Goto

Metodologia

1. Memória Compartilhada

1.1 Criando uma área para a memória compartilhada:

Para isso, utilizamos o comando *shm_open,* em que cria e mapeia de acordo com os parâmetros informados na chamada:

```
fd = shm_open("/sharedmemory", O_RDWR | O_CREAT, S_IRUSR | S_IWUSR);
if (fd == -1)
{
   perror("shm_open");
   exit(1);
}
```

1.2 Ajustando o tamanho da área compartilhada:

Ao chamarmos *ftruncate,* apenas informamos o descritor, retornado no passo anterior, e o tamanho do segmento desejado:

```
if (ftruncate(fd, sizeof(value)) == -1)
{
   perror("ftruncate");
   exit(1);
}
```

obs: declaramos anteriormente *value* como *double.*

1.3 Mapeando o seguimento de memória criado:

Primeiramente, criamos um ponteiro *“double *ptr;”* e utilizamos o *mmap* da seguinte forma:

```
ptr = mmap(NULL, SIZE, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
if (ptr == MAP_FAILED)
{
   perror("mmap");
   exit(1);
}
```

Em que o último parâmetro indica o ponto de início do mapeamento do arquivo e *SIZE* (definido anteriormente) o tamanho em bytes:

2. Processos

Como vimos anteriormente, ao utilizarmos *fork(),* criamos um novo processo idêntico ao pai no momento da criação.

Dessa forma, com o auxílio de uma variável global, iniciamos um loop, em que a cada iteração…

a) Move o ponteiro,
b) Chama *pi(i)* salvando sua parcela calculada na nova posição do ponteiro,
c) Cria um novo processo.

```
for (int i = 0; i < n_process; i++)
{
   ptr += sizeof(double);                          // Move o ponteiro.
   (*ptr) = pi(i);
   new_process = fork();                           // Cria um novo processo.

   if (new_process == 0)
   {
      printf ("Sou o %5d filho de %5d ", getpid(),getppid());
      break;
   }
}
```

Obs: *n_process* é informado pelo usuário.

```
double pi(int id) {
   long ID = (long)id;
        double sum = 0.0,x;
        unsigned long long int i;
   unsigned long long int passos = 1000000000;        // 1 Bilhão.
        double h=1.0/passos;

        for (i = ID + 1; i <= passos; i += n_process) {
                x = h * ((double)i - 0.5);
                sum += 4.0 / (1.0 + x*x);
        }
        resultados[ID] = h*sum;
   return resultados[ID];
}
```

3. Valor de Pi

Finalmente, criamos um novo laço, em que somamos cada parcela calculada pelos processos filhos, finalizados, em uma única variável, enquanto movemos o ponteiro para a próxima posição adequada, com o auxílio de *sizeof(double)*.

```c
for (int i = 0; i < n_process; i++) {
   printf("O processo %d terminou e ", wait(0));
   ptr += sizeof(double);                        // Move o ponteiro.
   value = (*ptr);
   printf("o pai leu o valor: %f\n", value);
   sum += value;
}
```

Resultado

Para compilação devemos utilizar:
gcc -Wall shm.c -o shm -lrt

Para iniciarmos devemos utilizar:
*./shm **x***
em que **x** deve ser um valor inteiro, representando a quantidade de processos desejados.

Ex: com 2 processos.

```
marcelo@marcelo-800G5M-800G5W:~/Área de Trabalho/Cursos/Sistemas Operacionais/TP
07/pi_processos_threads/Codigo$ gcc -Wall shm.c -o shm -lrt
marcelo@marcelo-800G5M-800G5W:~/Área de Trabalho/Cursos/Sistemas Operacionais/TP
07/pi_processos_threads/Codigo$ ./shm 2
Ola, sou o processo pai 15022

Sou o 15024 filho de 15022 e tenho o valor 1.570796
Sou o 15025 filho de 15022 e tenho o valor 1.570796
O processo 15024 terminou e o pai leu o valor: 1.570796
O processo 15025 terminou e o pai leu o valor: 1.570796

Valor de PI é: 3.141593
```

Ex: com 5 processos.

```
marcelo@marcelo-800G5M-800G5W:~/Área de Trabalho/Cursos/Sistemas Operacionais/TP
07/pi_processos_threads/Codigo$ ./shm 5
Ola, sou o processo pai 15079

Sou o 15080 filho de 15079 e tenho o valor 0.628319
Sou o 15081 filho de 15079 e tenho o valor 0.628319
Sou o 15082 filho de 15079 e tenho o valor 0.628319
Sou o 15083 filho de 15079 e tenho o valor 0.628319
Sou o 15084 filho de 15079 e tenho o valor 0.628319
O processo 15080 terminou e o pai leu o valor: 0.628319
O processo 15081 terminou e o pai leu o valor: 0.628319
O processo 15082 terminou e o pai leu o valor: 0.628319
O processo 15083 terminou e o pai leu o valor: 0.628319
O processo 15084 terminou e o pai leu o valor: 0.628319

Valor de PI é: 3.141593
```