Relatório Experimento 3
Sistemas Operacionais 1 – Turma 3 Grupo 8

Professor Tobar

Bruno de André Mazzoco RA:02150522
César Henrique Kallas RA: 02099224

Introdução

O experimento visa entender e implementar o uso e mecanismo de semáforos, uma implementação de gerenciamento de processos que garante o sincronismo entre dois processos concorrentes que disputam o mesmo recurso, como por exemplo uma variável, garantindo a exclusão mútua entre esses processos concorrentes, ou seja, somente um deles pode acessar tal recurso em um determinado tempo.

Programa Exemplo

O programa exemplo demonstrará o acesso simultâneo de três filhos a um mesmo recurso, de modo que cada um tentará acessar o maior número de vezes esse recurso.

Inicialmente são criadas duas estruturas, uma de memória compartilhada, recurso o qual os filhos irão requisitar, e um semáfaro.

Após isso os três filhos são criados e começam a requisitar a memória compartilhada (um vetor com letras do alfabeto) de modo a imprimirem o conteúdo desse recurso. O fato que pode acontecer, é que os filhos irão imprimir o alfabeto errado (fora de sequência), por isso o programa exemplo pode ser implementado de duas maneiras, essa primeira que mostramos e a segunda.

A segunda maneira, os três filhos irão fazer o acesso ao alfabeto, porém de maneira singular, um por vez ou seja, cada um deles tranca o semáforo, permanece em um loop um número de vezes variável, escrevendo um único caractere na tela a cada iteração, dormindo 1 ms para dar chance aos irmãos de executarem, copiando o índice atual de volta ao segmento de memória e verificando se o índice não ultrapassou o limite da string que vai ser impressa (se for ultrapassado ele escreve “/n”). Daí então, ele destrava o semáforo se for especificado pela implementação.

Resultados programa exemplo

COM PROTECT definido

abcdefghijklmnopqrstuvwxyz 1234567890 ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 1234567890 ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 1234567890 ABCDEFGHIJKLMNOPQRSTUVWXYZ

SEM PROTECT definido

a
a
abbbcccdddeeefffgggghhhiiijjjkkkllllmmmnnnoooppppqqqrrrsssttttuuuvvvwwwxxxy
yyzzz 1112223334445556667777888999000
AAABBBCCCDDDEEEFFFGGGHHHIIIJJJKKKLLLMMMNNNOOOPPPQQQRRRSSSTTTU
UUVVVWWWXXXYYYZZZ
a
a
abbbcccdddeeefffgggghhhiiijjjkkkllllmmmnnnoooppppqqqrrrsssttttuuuvvvwwwxxxy
yyzzz 1112223334445556667777888999000
AAABBBCCCDDDEEEFFFGGGHHHIIIJJJKKKLLLMMMNNNOOOPPPQQQRRRSSSTTTU
UUVVVWWWXXXYYYZZZ

Programa Criado

Tarefas:

- Crie um conjunto de semáforos com um único semáforo.
- Chame o relógio clock_gettime () para adquirir o valor da hora inicial.
- Chame semop () para trancar o semáforo.
- Chame semop () para destrancar o semáforo.
- Chame clock_gettime (), para adquirir o valor da hora final.
- Totalize as durações medidas e salve a maior e a menor delas.
- Repete os cinco passos anteriores

Resultados do Programa Criado

Número de programas exemplo*	Tempo máximo(s)	Tempo médio(s)
0	0.000004000000	0.000002652011
1	0.000004000000	0.000002652011
2	0.000004000000	0.000002656011
3	0.000004000000	0.000002654011
4	0.000004000000	0.000002652011
5	0.000004000000	0.000002658011
6	0.000004000000	0.000002654011
7	0.004004999995	0.000010693938
8	0.000004000000	0.000002656011
9	0.000004000000	0.000002652011

*programa exemplo – cutime.c

int main(int argc, char *argv)

```
{
   int x=1;

   while(x!=0)
   {
      x=x*x/x;
      x=x+5;
      x=x-5;
   }
}
```

Conclusão

Podemos notar que o tempo médio não se alterou muito, apenas em um momento, de certo porque o sistema estava ocupado no momento da syscall. O tempo médio é uma medida que corresponde a quanto tempo, em media, demorou para que o SO travasse e depois destravasse o semáforo durante 500 iterações, não deixando que a seção dita “critica” fosse utilizada por um outro processo. Isso significa que na média, foi garantida a exclusão mutua por um período bem pequeno porem suficiente para que nenhum outro processo utilizasse a seção crítica ao mesmo tempo. Tal medida sendo pequena serve para demonstrar a eficiência da utilização de semáforos para garantir a exclusão mútua, o que para o SO significa melhor gerencia entre processos e otimização num ambiente multi-tarefas. Alem de ser mais eficiente do que outros modos de sincronização, o uso de semáforos é de fácil implementação. A medida do tempo Maximo serve para mostrar o valor de tempo em que se demorou mais para se executar o mecanismo de travar e destravar o semáforo durante um período de 500 iterações, o que no caso do experimento não foi muito maior do que a media obtida (pelo menos mesma ordem de grandeza). A todas essas medidas, podemos atribuir um pequeno erro na tomada dos tempos pela execução de alguns comandos de gerenciamento de memória compartilhada (comandos “IF”) sendo esses tempos desconsiderados na contagem.