

Relatório Experimento 3

Sistemas Operacionais 1 – Turma 3 Grupo 8

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

Introdução

Este experimento visa o entendimento de mais um método de comunicação entre processos concorrentes(IPC) através da utilização de memória compartilhada (região da memória onde vários processos podem acessar). O experimento objetiva, assim, entender o que são sinais e como eles são usados, além de observar as limitações e dificuldades em seu uso e também demonstrar a utilização de manipuladores de sinais e criação de mascaras para os sinais.

Programa Exemplo

O programa exemplo basicamente faz uma medição de tempo da latência entre o envio e recebimento de um sinal mandado de um processo para outro, através da memória compartilhada. Para isso, é definida uma estrutura que guarda os tempos a serem medidos, é criado um ponteiro do mesmo tipo da estrutura definida, cria-se e associa-se um segmento de memória compartilhada sendo que tal associação retorna um endereço de memória que é armazenado no ponteiro previamente criado. São criados os dois filhos que irão se comunicar que herdam uma mesma mascara de sinais do pai. Os filhos criarão um conjunto de sinais e estabelecerão então o manipulador de sinais. Eles criam uma mascara do conjunto de sinais já existente, adicionam um sinal à mesma que contem também o numero do processo que ira mandar o sinal. Tal manipulador só poderá ser utilizado por um processo por vez e os sinais são bloqueados e só são entregues quando os mesmos são desbloqueados, ou seja, quando nenhum outro processo esta utilizando a mascara de sinais. O calculo do tempo é feito em um loop que executa a medição um numero de determinado de vezes , onde um filho toma os tempos do primeiro campo da matriz de tempo (definida dentro da estrutura primeiramente criada) e o outro filho toma os tempos do segundo campo da matriz de tempo.

Resultados Programa Exemplo

Número da execução	Número de programas exemplo*	Latência máxima(ms)	Latência média (ms)
1	0	0.001622	0.000007
2	1	0.000208	0.000007
3	3	0.000882	0.000007
4	5	0.000190	0.000006
5	7	0.000251	0.000007
6	9	0.000205	0.000007
7	11	0.000434	0.000007
8	13	0.001322	0.000009
9	15	0.001219	0.000009
10	17	0.000966	0.000010

- programa exemplo cutime.c, rodando para ocupar a cpu.

```
#include <stdio.h>
```

```
int main(int argc, char *argv)
{
    int x=1;

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

Modificação do Programa Exemplo

Foram feitas as seguintes modificações do programa exemplo:

- Ambos os filhos continuam fazendo o mesmo porem quando utilizam o manipulador terão que dormir um determinado tempo lido pelo teclado entre a tomada de tempo e envio de outro sinal

- A cada 200 interações será apresentado o valor total dos tempos e a media dos tempos ate aquele momento , apresentando- se também o PID de cada processo;

Resultados do Programa Modificado

Número da execução	Número de programas exemplo*	Atroso em milisegundos	Latência máxima(s)	Latência média (s)
1	0	0	0.007229	0.000023
2	1	20	0.089957	0.010070
3	3	40	0.043029	0.010016
4	5	60	0.060238	0.010020
5	7	80	0.035893	0.010000
6	9	100	0.050196	0.010030
7	11	120	0.041313	0.010010
8	13	140	0.060474	0.010030
9	15	160	0.059160	0.010045
10	17	180	0.037631	0.010000
11	20	2000	0.036815	0.010000

*número de programas exemplo de tomada de cpu rodando em conjunto

Análise dos resultados

Os resultados são muito intrigantes, como na maioria dos exemplos anteriores.

Não há uma seqüência lógica para demandar o crescimento dos tempos. Se por um lado aumentamos a taxa de processamento, com mais carga, os tempos correspondem aumentando, mas nem sempre. Isso corresponde claro ao fato de que, a medição de tempo não é real, em melhores palavras, não é possível saber exatamente o tempo gasto por um processo apenas medindo o tempo de partida e chegada nos moldes de sistemas operacionais que utilizamos hoje em dia.

No programa modificado houve um fato muito interessante, na partida 11, aumentou- se o tempo de atraso em muitas unidades a mais do que na partida anterior, porém o resultado de tomadas médias foi o mesmo, e a latência máxima foi menor.

Há alguma explicação ? Sim e várias, acredito que a mais significantes seja o tempo de processamento, pois na tomada 11 não deve ter tido interrupção (executando <- > prontos).

Conclusão:

Podemos concluir que a troca de mensagens por memória compartilhada é muito mais rápida do que por filas, porém tem seus lados negativos, como a forte proteção a área de seu acesso.

Se houver uma necessidade de troca de mensagens e que a mesma seja a mais precisa em termos de tempo (mais rápida), com certeza o melhor método até agora estudado por nós é esse.

O entendimento do programa exemplo desse experimento foi mais complicado, porém na prática ele se torna mais simples do que parece, pois ao entender os conceitos, sua implementação fica tranquila. Claro que entender os conceitos nem sempre é uma regra de facilitação para a prática.