**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS – PUC-CAMPINAS**

***Experimento 2***

***Sistemas Operacionais A***

|  |  |
| --- | --- |
| **ALUNO** | **RA** |
|
| Beatriz Morelatto Lorente | 18071597 |
| Cesar Marrote Manzano | 18051755 |
| Fabricio Silva Cardoso | 18023481 |
| Pedro Ignácio Trevisan | 18016568 |

**-= Sumário =-**

**1 Introdução........................................................................................................................3**

**2 Apresentação dos erros do programa exemplo e suas soluções..............................**

**3 Resultados da execução do programa exemplo..........................................................**

**4 Resultados da execução do programa modificado.....................................................**

**5 Respostas das perguntas.............................................................................................**

**6 Análise dos Resultados................................................................................................**

**7 Conclusão......................................................................................................................**

**-= Introdução =-**

O experimento realizado permitiu o entendimento da comunicação entre processos ou IPC (Inter-Process Communication). O experimento foi dividido em duas tarefas, com o objetivo de mostrar os tempos de transferência entre mensagens (enviadas através do mecanismo de fila de mensagens).

Na primeira tarefa, foi executado um programa simples, no qual um processo filho (Sender) mandava uma mensagem para outro (Receiver). O Receiver calculava o tempo médio e máximo de transferência e imprimia as informações.

Na segunda tarefa, havia três processos filhos, sendo que um era responsável por mandar as informações (Sender) para um outro processo (Receiver1) e este por um breve instante mandava uma mensagem para outro filho (Receiver2). O Receiver1 calculava o tempo total, médio, máximo e mínimo de transferência e envia essas informações para o Receiver2 imprimir. O usuário também poderia escolher o tamanho da mensagem que seria enviada, sendo que o usuário escolhia um número de 1 a 10, e este era multiplicado por 512, (representando 512kB).

**-= Apresentação dos erros do programa exemplo e suas soluções =-**

**-= Resultados da execução do programa exemplo =-**

**-= Resultados da execução do programa modificado =-**

**-= Respostas das perguntas =-**

**Perguntas do relatório**

***Pergunta 1:* Esclarecer o que são: Berkeley Unix, System V, POSIX, AT&T, socket, fila de mensagem, memória compartilhada e pipes.**

***Pergunta 2: As chamadas ipcs e ipcrm apresentam informações sobre quais tipos de recursos?***

***Pergunta 3: Qual a diferença entre o handle devolvido pela chamada msgget e a chave única?***

***Pergunta 4: Há tamanhos máximos para uma mensagem? Quais?***

***Pergunta 5: Há tamanhos máximos para uma fila de mensagens? Quais?***

***Pergunta 6: Para que serve um typedef?***

***Pergunta 7: Onde deve ser usado o que é definido através de um typedef?***

***Pergunta 8: : Na chamada msgsnd há o uso de cast, porém agora utiliza-se um “&” antes de message\_buffer. Explicar para que serve o “&” e o que ocorreria se este fosse removido.***

**Perguntas do programa**

***Pergunta 1: O que é um protótipo? Por qual motivo é usado?***

***Pergunta 2: O que significa cada um dos dígitos 0666?***

***Pergunta 3: Para que serve o arquivo stderr?***

***Pergunta 4: Caso seja executada a chamada fprintf com o handler stderr , onde aparecerá o seu resultado?***

***Pergunta 5: Onde stderr foi declarado?***

***Pergunta 6: Explicar o que são e para que servem stdin e stdout.***

***Pergunta 7: O que ocorre com a fila de mensagem se ela não é removida e os processos terminam?***

***Pergunta 8: Qual será o conteúdo de data\_ptr?***

**-= Análise dos Resultados =-**

**-= Conclusão =-**