

INF1316 – Sistemas Operacionais – Lab Semáforos

- 1) Faça um programa que utilize a memória compartilhada para trocar 128 mensagens entre 2 processos/programas independentes. Use semáforos para sincronizar a aplicação da seguinte forma:

- a) Processos síncronos:

Processo 1 escreve mensagem na memória compartilhada, processo 2 lê a mensagem da memória compartilhada e informa ao processo 1 que pode enviar nova mensagem. As mensagens são strings de, no máximo, 15 caracteres e têm o seguinte formato:

“mensagem 1”,

“mensagem 2”, etc.

As mensagens são tratadas uma de cada vez.

- b) Processos assíncronos:

Processo 1 escreve as mensagens para serem tratadas pelo processo 2. O buffer de comunicação entre os processos, que fica na memória compartilhada, é capaz de armazenar apenas 8 mensagens. As mensagens são strings e são exibidas a fim de mostrar o que está ocorrendo na aplicação. Deve ser informado ainda que a mensagem foi enviada pelo processo x e que foi recebida pelo processo y. Os processos devem, obrigatoriamente, ser concorrentes. As mensagens são strings de, no máximo, 15 caracteres e têm o seguinte formato:

“mensagem 1”,

“mensagem 2”, etc.

Entregue os códigos fonte dos 2 exercícios (1a e 1b) juntamente com o relatório que exhibe quais foram os resultados obtidos, explique como foi garantida a concorrência e como foram implementados os sincronismos entre os processos/programas. Explique ainda o que funciona e o que não funciona no seu programa.

Para usar semáforos use a biblioteca “semaphore.h”

Chamadas:

var= **sem_open(“nome semáforo”, O_CREAT, 0666, valor inicial),**

Onde: O_CREAT está definido em <fcntl.h>

0666 é a permissão de uso do semáforo

valor inicial=0

OU

var= **sem_init(...)**

E as operações:

Down: **sem_wait(var)**

Up: **sem_post(var)**

Close: **sem_close(var)**

Elimina semáforo: **sem_unlink(nome_semáforo)**