

Relatório do Laboratório adicional de INF 1036 - Sistemas Operacionais

Laboratório Adicional - memória compartilhada

Data: 20/09/2023

Alunos:

Leo Lomardo - matrícula 2020201

Lucas Lucena - matrícula 2010796

Objetivo:

Criar dois programas que compartilham de uma mesma memória, no qual o primeiro cria a memória com uma chave específica, recebe um input do usuário contendo uma frase do dia, e salva essa frase na memória. O segundo programa acessa essa memória e mostra ao usuário qual a frase do dia.

Estrutura dos programas:

Programa 1 - Criação da Memória Compartilhada - cria_mem.c

Declaração das bibliotecas necessárias

Declaração das variáveis: São declaradas variáveis para a chave (key_t chave), o identificador do segmento de memória compartilhada (int segmento), um ponteiro para a área de memória compartilhada (char *p), e um array para armazenar a mensagem do dia (char texto[100]).

Leitura da mensagem: O programa solicita que o usuário insira a mensagem do dia, que é lida a partir da entrada padrão usando a função fgets e armazenada na variável texto.

Criação da memória compartilhada: A memória compartilhada é criada com a chave 7000 usando a função shmget. Em caso de erro, o programa exibe uma mensagem de erro e encerra.

Anexação à memória compartilhada: A memória compartilhada é anexada ao processo com a função shmat. Em caso de erro, uma mensagem de erro é exibida e o programa encerra.

Cópia da mensagem: A mensagem do dia fornecida pelo usuário é copiada para a memória compartilhada usando strcpy.

Conclusão: Uma mensagem de confirmação é exibida, informando que a mensagem foi salva na memória compartilhada.

Programa 2 - Acesso à Memória Compartilhada - le_mem.c

Declaração das bibliotecas necessárias

Declaração das variáveis: São declaradas variáveis para a chave (key_t chave), o identificador do segmento de memória compartilhada (int segmento), e um ponteiro para a área de memória compartilhada (char *p).

Obtenção da memória compartilhada: O programa tenta obter a memória compartilhada existente com a chave 7000 usando a função shmget. Em caso de erro, uma mensagem de erro é exibida e o programa encerra.

Anexação à memória compartilhada: A memória compartilhada é anexada ao processo com a função shmat. Em caso de erro, uma mensagem de erro é exibida e o programa encerra.

Exibição da mensagem: A mensagem do dia, que está armazenada na memória compartilhada, é exibida usando printf.

Liberação da memória compartilhada: O programa libera a memória compartilhada do processo com shmdt e, em seguida, libera a memória compartilhada definitivamente com shmctl.

Solução:

O programa cria_mem realiza os seguintes passos:

Solicita ao usuário que insira a mensagem do dia.

Cria uma memória compartilhada com a chave 7000 usando shmget.

Anexa a memória compartilhada ao processo com shmat.

Copia a mensagem do usuário para a memória compartilhada com strcpy.

Exibe uma mensagem de confirmação.

Código:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/shm.h>

int main() {
    key_t chave = 7000;
    int segmento;
```

```

char *p;
char texto[100];

// Lê da entrada a mensagem do dia
printf("Digite a mensagem do dia: ");
fgets(texto, sizeof(texto), stdin);

// Cria a memória compartilhada com a chave 7000
segmento = shmget(chave, 1024, IPC_CREAT);
if (segmento == -1) {
    perror("Erro ao criar a memória compartilhada");
    exit(1);
}

// Anexa a memória compartilhada ao processo
p = (char *)shmat(segmento, NULL, 0);
if (p == (char *)-1) {
    perror("Erro ao anexar a memória compartilhada");
    exit(1);
}

// Copia a mensagem para a memória compartilhada
strcpy(p, texto);
printf("Mensagem salva na memória compartilhada.\n");

return 0;
}

```

O programa le_mem realiza os seguintes passos:

- Obtém a memória compartilhada existente com a chave 7000.
- Anexa a memória compartilhada ao processo.
- Exibe a mensagem do dia armazenada na memória compartilhada.
- Libera a memória compartilhada do processo com shmdt.
- Libera a memória compartilhada definitivamente com shmctl.

Código:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>

int main() {
    key_t chave = 7000;
    int segmento;
    char *p;

    // Obtém a memória compartilhada existente
    segmento = shmget(chave, 0, 0);
    if (segmento == -1) {
        perror("Erro ao obter a memória compartilhada");
        exit(1);
    }

    // Anexa a memória compartilhada ao processo
    p = (char *)shmat(segmento, 0, 0);
    if (p == (char *)-1) {
        perror("Erro ao anexar a memória compartilhada");
        exit(1);
    }

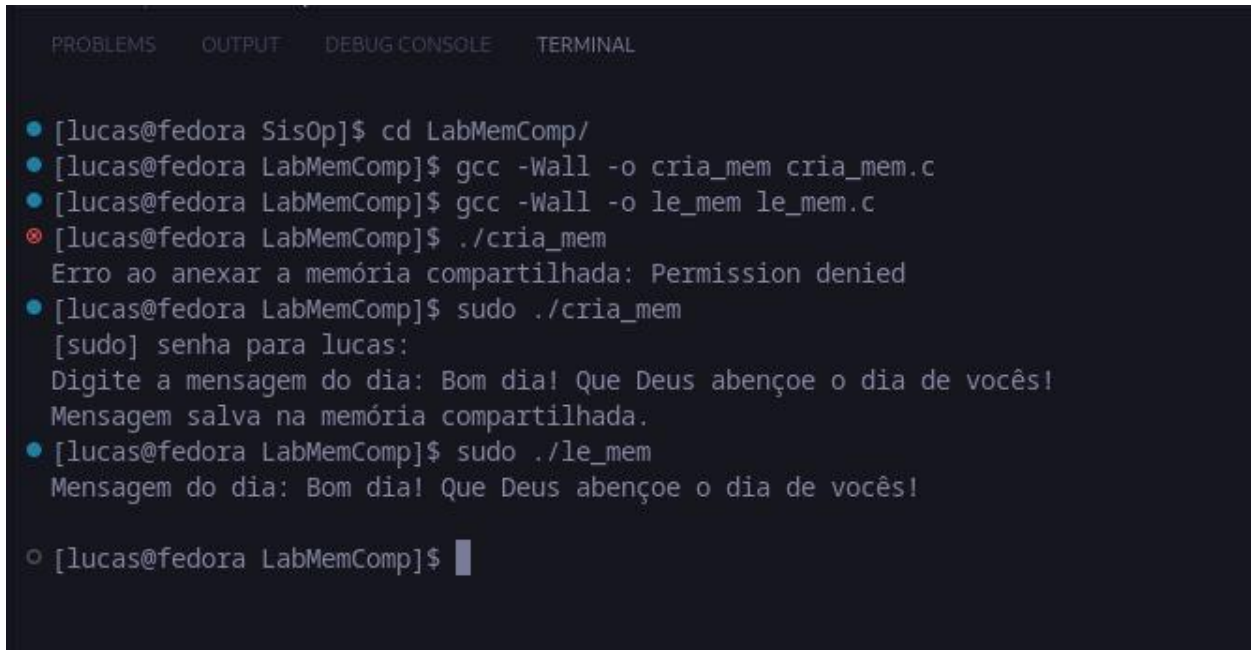
    printf("Mensagem do dia: %s\n", p);

    // Libera a memória compartilhada do processo
    shmdt(p);

    // Libera a memória compartilhada
    shmctl(segmento, IPC_RMID, 0);

    return 0;
}
```

Observações e conclusões:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

• [lucas@fedora SisOp]$ cd LabMemComp/
• [lucas@fedora LabMemComp]$ gcc -Wall -o cria_mem cria_mem.c
• [lucas@fedora LabMemComp]$ gcc -Wall -o le_mem le_mem.c
✖ [lucas@fedora LabMemComp]$ ./cria_mem
Erro ao anexar a memória compartilhada: Permission denied
• [lucas@fedora LabMemComp]$ sudo ./cria_mem
[sudo] senha para lucas:
Digite a mensagem do dia: Bom dia! Que Deus abençoe o dia de vocês!
Mensagem salva na memória compartilhada.
• [lucas@fedora LabMemComp]$ sudo ./le_mem
Mensagem do dia: Bom dia! Que Deus abençoe o dia de vocês!

○ [lucas@fedora LabMemComp]$
```

O código funciona perfeitamente, porém somente se for executado com o comando sudo. Nos primeiro testes, não foi necessária a permissão de superusuário, porém, a partir de certo ponto, foi.