

Sistemas Operacionais

Experimento #5

1. Introdução

O IPC (*Inter-Process Communication*), ou comunicação entre processos, consiste de um conjunto de métodos que permitem que processos no Sistema Operacional Unix possam se comunicar. Em um dos experimentos anteriores foram exploradas as chamadas para alguns mecanismos de IPC existentes no System V, mais especificamente as filas de mensagens.

Neste experimento, deseja-se que filas de mensagens sejam usadas para propiciar sincronismo no problema do Barbeiro Dorminhoco. Deseja-se também que a solução desse problema seja implementada com *threads*.

Este exercício foi inspirado pelos experimentos existentes em <http://www.rt.db.erau.edu/experiments/unix-rt> que pertencem ao Laboratório Embry-Riddle de tempo-real.

2. Objetivos

A seguir estão os objetivos deste experimento com relação ao aluno:

- Rever o problema clássico do Barbeiro Dorminhoco.
- Utilizar o mecanismo de troca de mensagens para obtenção de sincronização.
- Rever o mecanismo de *semáforo*.
- Rever o mecanismo de memória compartilhada.
- Rever e reforçar o aprendizado do mecanismo de troca de mensagens.
- Rever o mecanismo de *thread*.
- Rever o mecanismo de *mutex*.

3. Tarefas

Deseja-se a implementação do algoritmo do Barbeiro Dorminhoco, em processos, utilizando-se como mecanismo de sincronismo a troca de mensagens, na linguagem C, como se segue:

- Substitua cada um dos semáforos barbers e customers (solução logo a seguir) por uma única fila de mensagens.
- Inicialize dois processos barbeiros e 20 processos clientes (nessa ordem - o número de cadeiras deve ser sete).

Pergunta 1: qual é o recurso comum que necessita de exclusão mútua?

Pergunta 2: de que maneira (leitura, escrita, ambos) barbeiros e clientes vão acessar o recurso comum?

- Como parte do resultado deseja-se saber se algum dos clientes não foi atendido.

- Um cliente deve passar ao barbeiro que lhe atende, na mensagem, um *string* com um número variável de números inteiros, aleatoriamente definidos entre 2 (dois) e 1023 (mil e vinte e três).

Pergunta 3: como os números foram colocados no *string*?

- Em `cut_hair(i)` converta o *string* enviado pelo cliente em vários inteiros. Ordene esses números, de forma decrescente.
- O barbeiro devolve uma mensagem ao respectivo cliente com o resultado do serviço prestado.
- Em `apreciate_hair()`, exiba o número do cliente, o número do barbeiro, o conteúdo do que se quer ordenar, o resultado da ordenação e o tempo de demora para ser atendido (desde o instante que entra e senta na sala de espera, até o momento em que começa o corte do cabelo (início de `cut_hair()`)).

Deseja-se também uma segunda implementação do algoritmo do Barbeiro Dorminhoco usando:

- *threads*, ao invés de processos.
- *mutex*, ao invés de semáforo para a exclusão mútua.
- Semáforos `customers` e `barbers`, ao invés de fila de mensagens.
- Inicialize três processos barbeiros e 27 processos clientes (mantenha o número de cadeiras em sete).
- O corte de cabelo dos clientes significa uma ordenação de forma crescente.
- Não se deve usar fila de mensagens.

Pergunta 4: Como o barbeiro vai ter acesso aos valores a serem ordenados?

Pergunta 5: Como o cliente vai ter acesso aos resultados?

- O cliente deve ter acesso e exibir seu número, o número do barbeiro que o atendeu e quanto tempo demorou na barbearia (desde o momento que entrou até o início de `apreciate_hair()`), além dos números que se quer ordenar e o resultado do serviço.
- Os clientes que não foram atendidos devem continuar sua execução, por exemplo indo tomar um sorvete, e só terminar depois de ter seu cabelo cortado.

4. Teoria

No experimento #2 encontra-se a teoria relativa ao mecanismo de troca de mensagens. No livro texto encontra-se o algoritmo do Barbeiro Dorminhoco. No experimento #3 encontra-se a teoria relativa a memória compartilhada e a semáforos. No experimento #4 encontra-se a teoria relativa aos mecanismos de *threads* e *mutex*.

5. Resultado

Este experimento, diferente dos demais, não contempla um programa exemplo para ser compilado, embora a atividade a ser desenvolvida considere um algoritmo existente no

livro usado como referência (Tanenbaum, 2010). Considere que *down* é sinônimo de P e que *up* é sinônimo de V.

```
# define CHAIRS = 5;
semaphore exc_aces, customers, barbers;
shared int waiting;
void barber(int i)
{
    while (1 == 1) {
        down(&customers);
        down(&exc_aces);
        waiting=waiting-1;
        up(&exc_aces);
        cut_hair(i);
        up(barbers);
    }
}

void customer(int i)
{
    down(&exc_aces);
    if waiting < CHAIRS
    {
        waiting=waiting+1;
        up(&customers);
        up(&exc_aces);
        down(&barbers);
        appreciate_hair();
    }
    else
        up(&exc_aces);
}

int main(int argc, char *argv[])
{
    init(exc_aces,1);
    init(customers,0);
    init(barbers,0);
    waiting=0;
    parbegin
        barber(1); barber(2); barber(3);
        customer(1); ... customer(20);
    parend;
}
```

Este experimento deve ser acompanhado de um relatório com as seguintes partes obrigatórias:

- Introdução, indicando em não mais do que 20 linhas o que fazem o programa exemplo e os programas modificados;
- Respostas às perguntas que se encontram dispersas pelo texto do experimento e pelo código fonte exemplo;
- Resultados da execução dos programas;

- Análise dos resultados;
- Conclusão indicando o que foi aprendido com o experimento.

Entrega

A entrega do experimento deve ser feita em seu escaninho no AVA, em uma pasta com o nome “Experimento5”, de acordo com o cronograma previamente estabelecido, **até a meia-noite** do dia anterior à respectiva apresentação.

Em todos os arquivos entregues deve constar **OBRIGATORIAMENTE** o nome e o RA dos integrantes do grupo.

Devem ser entregues os seguintes itens:

- i. *os códigos fonte*;
- ii. os executáveis;
- iii. o relatório final do trabalho, em formato pdf.

Solicita-se que **NÃO** sejam usados compactadores de arquivos.

Não serão aceitas entregas após a data definida. A não entrega acarreta em nota zero no experimento.

6. Apresentação

Os resultados do experimento serão apresentados em sala no dia de aula prática da semana marcada para a entrega, com a presença obrigatória de todos os alunos, de acordo com o cronograma previamente estabelecido.

Serão escolhidos alunos para a apresentação e discussão dos resultados.

Todos os alunos que completaram o experimento devem preparar para a apresentação, em formato digital:

- Os códigos-fonte,
- A introdução e a conclusão,
- As respostas às questões,
- Os resultados e sua análise.

Recomenda-se fortemente a preparação para apresentação.