



Universidade Federal do Ceará – UFC

Lista 2 – Engenharia da Computação

Disciplina: Sistemas Operacionais

Professor: Joniel Bastos

Nome: _____ **N:** _____

Parte 1 – Threads

1. Crie um programa em C que gere um processo com 100 threads e execute a seguinte ação:
 - O processo pai escreve na tela logo após ser criado "Alô do pai" e em seu último comando escreve "Tchau do pai!"
 - As threads filhas escrevem na tela logo após ser criado "Alô do filho [1]" e em seu último comando escreve "Tchau do filho [1]!"
 - Coloque entre os comandos de escrita alguns comandos para fazer com que os processos demorem um pouco mais.

2. Podemos calcular o seno de um número segundo a série de Taylor-Maclaurin:

$$\text{sen}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

Faça um programa em C que lê um valor para x e calcule o valor de Sen(x) utilizando duas threads para dividir os cálculos, uma para os termos com soma e outra para os com subtração. O valor deve ser calculado até ser próximo do seno.

3. Faça um programa que leia um número “n” informado pelo usuário e diga quantos números primos há entre 0 e “n”. Esse seu programa deve rodar em 2 threads.

Parte 2 – Escalonamento

Implemente uma simulação para os algoritmos de escalonamento FIFO, SJF e Round Robin em C. Em sua implementação o usuário deve fornecer a quantidade de processos que estão na fila para serem executados, a ordem de chegada e o tempo de execução de cada processo. Durante a execução deve ser informado ao usuário qual o processo que está executando e esperado o tempo de cada processo por segundo. Por exemplo, se um processo A demora 5 segundos, deve ser informado que ele está sendo trabalhado e esperado o tempo de execução deste para poder iniciar o próximo, para isso utilize um temporizador em seu algoritmo.

Obs: Escrever um relatório explicando toda a carga teórica dos algoritmos implementados nos programas, explicando o código e a lógica utilizada. O relatório e os códigos devem ser enviados em uma pasta compactada.

Parte 3 – Sistema de arquivos

4. Escreva um programa que comece em um determinado diretório e percorra a árvore de arquivos a partir daquele ponto registrando os tamanhos de todos os arquivos que encontrar. Quando houver concluído, ele deve imprimir um histograma dos tamanhos dos arquivos usando uma largura de célula especificada como parâmetro (por exemplo, com 1024, tamanhos de arquivos de 0 a 1023 são colocados em uma célula, 1024 a 2047 na seguinte etc.).

5. Escreva uma nova versão do programa ls do UNIX. Essa versão recebe como argumentos um ou mais nomes de diretórios e para cada diretório lista todos os arquivos nele, uma linha por arquivo.