

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM INFORMÁTICA
SISTEMAS OPERACIONAIS I
2^o SEM/2011
Trabalho em Grupo – Nr 2

ESTUDO SOBRE SUBPROCESSOS COOPERATIVOS

1. Objetivo do Trabalho

Estimular a capacidade do aluno de trabalhar em equipe para organizar, projetar e desenvolver soluções para problemas formulados que envolvam o estudo e o conhecimento sobre subprocessos e threads.

2. Escopo do Trabalho

- ✓ Estudar comandos indicados.
- ✓ Conceber e implementar os algoritmos conforme as questões apresentadas.
- ✓ Preparar um relatório em Word.
- ✓ Entregar todo o material elaborado (códigos fontes, executáveis e relatório) em meio ótico (CDROM). O relatório, além de constar do CD, deve também ser entregue impresso.
- ✓ Incluir no relatório um extrato da console de execução dos programas.

3. Equipes de Trabalho

Devem ser formadas com 3 alunos cada. Excepcionalmente pode haver uma equipe com 2 alunos tendo em vista o número de inscritos.

4. Prazo de Entrega do Trabalho

O material deverá ser entregue na aula do dia **21/09**.

5. Penalidades

Caso o grupo atrase a entrega do resumo seu grau final sofrerá um decréscimo na razão de 0,5 pontos por dia.

6. Avaliação

Serão considerados os seguintes aspectos:

- ✓ Estética da apresentação do relatório e seu conteúdo;
- ✓ Execução correta dos programas.

7. Temas para Desenvolvimento

- a. Estudo de Comandos

Estude os comandos: `fork()`; `exec()`; `execl()`; `wait()` e `exit()`; `getpid()`, `getppdi()`.

Leia o material sobre Comunicação entre Processos.

b. Prog1 (anexo) - Uso dos comandos *fork()* e *exec()*

- ✓ Prepare o código do programa segundo os requisitos solicitados.
- ✓ Execute o programa e responda às questões lá colocadas no relatório.
- ✓ Mostre no relatório o conteúdo da console durante a execução.

c. Prog2 (anexo) – Árvore genealógica e escopo das variáveis

- ✓ Prepare o código do programa segundo os requisitos solicitados.
- ✓ Faça “m = 1”, execute o programa e responda às questões lá colocadas.
- ✓ Verifique a função da variável “j” e certifique-se de que funciona corretamente.
- ✓ Verifique se pode incluir no código uma forma dos processos pais armazenarem os PIDs de seus processos filhos e apresentar estes PIDs no momento em que o processo pai inicia a espera pelo término dos mesmos.
- ✓ Faça sucessivamente “m = 2” e “m = 3”, avalie a árvore genealógica que é gerada em cada caso e faça a previsão do número total de processos a serem ativados nos casos em que “m=4” e “m=10”.
- ✓ Verifique e apresente suas conclusões sobre a ordem em que os processos são ativados e a ordem em que ganham a CPU.

d. Prog3 – Subprocessos Cooperativos

Construa um programa que:

- a) Gere aleatoriamente, com números inteiros entre -100 e +100, uma matriz “A” de dimensões “m” linhas e “k” colunas (fornecidos como parâmetros em tempo de execução);
- b) Encontre e mostre o menor e o maior valor da matriz e suas respectivas coordenadas;
- c) Calcule os “m” produtos internos (linha x coluna) conforme a fórmula

$$PI_i = \sum_{j=1}^k A_{i,j} * A_{j,i}$$

- d) Encontre e mostre o maior, o menor e o desvio padrão dos produtos internos gerados e suas respectivas coordenadas.

(*) Obs:

1. Em todas as versões do programa, o tempo total de execução deve ser computado e apresentado na tela ao final da execução.
2. O programa deve permanecer em “loop” até que seja fornecido um valor zero para “m” e “k”.

3. Rode o programa em uma máquina que possua pelo menos dois processadores. Informe no relatório a configuração da máquina utilizada.

✓ **Versão 1 do programa:** Construa toda a solução em um único fluxo de execução.

✓ **Versão 2 do programa:** Faça uso de um subprocesso para calcular cada produto interno e avalie o desempenho do programa a medida que “m” aumenta.

Teste e compare o desempenho das duas versões para diferentes valores de “m” e “k” (varie-os de pequenos – da ordem de dezenas ($m = 10$ e 20 , $k = 20, 30, 100, 1000, 10000$ até $m = 10, 20, 100, 1000$ e 5000 e $k = 20, 30$ e 100 , por exemplo).

Analise os resultados de desempenho alcançados, mostre-os em uma tabela e apresente suas conclusões.

✓ **Versão 3 do programa:** Sugira e teste uma forma diferente de distribuir o trabalho entre os subprocessos de maneira a evitar possíveis problemas encontrados na versão 2 do código. Explique sua estratégia e apresente os resultados alcançados.

BOM TRABALHO

Prog1.c

```
#include <stdio.h>
#include <wait.h>
#include <unistd.h>

int main(void)
{
    int    status, id, j;

    ***** Insira um comando para pegar o PID do processo corrente e mostre na tela da
    console.

    if (*** insira um comando para criar um subprocesso)
    {
        ***** Faça com que o processo pai execute este trecho de código
        ***** Mostre na console o PID do processo pai e do processo filho
        ***** Monte uma mensagem e a envie para o processo filho
        ***** Mostre na tela o texto da mensagem enviada
        ***** Aguarde a resposta do processo filho
        ***** Mostre na tela o texto recebido do processo filho
        ***** Aguarde mensagem do filho e mostre o texto recebido
        ***** Aguarde o término do processo filho
        ***** Informe na tela que o filho terminou e que o processo pai também vai encer-
        rar
    } else
    {
        ***** Faça com que o processo filho execute este trecho de código
        ***** Mostre na tela o PID do processo corrente e do processo pai
        ***** Aguarde a mensagem do processo pai e ao receber mostre o texto na tela
        ***** Envie uma mensagem resposta ao pai
        ***** Execute o comando “for” abaixo
        for (j = 0; j <= 10000; i++);
        ***** Envie mensagem ao processo pai com o valor final de “j”
        ***** Execute o comando abaixo e responda às perguntas
        execl("/Bin/l", "ls", NULL);

        ***** O que acontece após este comando?
        ***** O que pode acontecer se o comando “execl” falhar?
    }
    exit(0);
}
```

Prog2.c

```
#include <stdio.h>
#include <wait.h>
#include <sys/types.h>

#define    m    k;

int        i, j, k, id, d1, d2, status;

int main(void)
{
    **** inicialize as variáveis d1 e d2 com valores distintos;
    **** mostre o PID do processo corrente e os valores de d1 e d2 na tela da console
    **** responda: quais processos executarão este trecho do código?

    j = 0;

    for (i = 0; i <= m; i++)

    {
        ***** mostre na tela da console, a cada passagem, os seguintes valores: PID do
        processo corrente; “i”, “d1”, “d2” e “m”

        ***** responda: quais processos executam este trecho do código?

        id = fork();

        if (id)
        {

            ***** altere os valores de d1 e d2 de diferentes maneiras como exemplificado
            abaixo

            d1 = d1 + (i + 1);

            d2 = d2 + d1 * 3;

            ***** mostre na tela da console, a cada passagem, os seguintes valores: PID do
            processo corrente; “i”, “d1”, “d2”, “m” e informe estar no ramo “then” do “if”

            ***** responda: quais processos executam este trecho do código?

        } else
        {

            ***** altere os valores de d1 e d2 de diferentes maneiras e também diferente
            do usado no trecho “then”

            ***** execute o comando de atualização de “j” abaixo

            j = i + 1;
```

```

        ***** mostre na tela da console, a cada passagem, os seguintes valores: PID do
        processo corrente; “i”, “d1”, “d2”, “m” e informe estar no ramo “then” do “if”

        ***** responda: quais processos executam este trecho do código?
    }
}

        ***** responda: quais processos executam este trecho do código?

if (id != 0)
{
    ***** mostre na console o PID do processo corrente e verifique quais proces-
    sos executam este trecho do código

    for (i = j; i == m; i++)
    {
        ***** explique o papel da variável “j” e verifique se o comando “for” está
        correto de forma a que cada processo pai aguarde pelo término de todos seus
        processos filhos

        ***** mostre na console o PID do processo corrente e o número de filhos
        que ele aguardou ou está aguardando

        wait(&status);

        if (status == 0)

            ***** responda: o que ocorre quando este trecho é executado?
        else

            ***** responda: o que ocorre quando este trecho é executado?

    }
}
exit(0);
}

```

Rio de Janeiro, 3 de setembro de 2011