DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
SISTEMAS OPERACIONAIS
2₀ SEM/2018
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 para todos os exercícios solicitados.
- ✓ Entregar todo o material elaborado (códigos fontes e relatório) por email.
- ✓ **Incluir no relatório a saída da console durante a execução dos programas**.

3. Equipes de Trabalho

Devem ser formadas com **até 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 **20/11/2018.**

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 durante a avaliação;
- ✓ Desenvoltura e conhecimento do grupo durante a apresentação.

7. Temas para Desenvolvimento

Utilize o ambiente Linux para desenvolver seus programas.

a. <u>Estudo de Comandos</u>

- ✓ Estude os comandos: fork(); exec(); execl(); wait() e exit(); getpid(), getppid(). Porém, não é necessário efetuar nenhum comentário a respeito no relatório.
- ✓ Leia o material sobre Comunicação entre Processos ou execute o comando "man" em ambiente Unix ou Linux.

b. <u>Anexo 1 – Exemplos de uso das funções fork() e exec()</u>

- ✓ Execute os exemplos dados no **Anexo 1**, e descreva no relatório o que será executado, apresentando os resultados obtidos.
- ✓ Para esses exemplos não será cobrada a apresentação.

c. <u>Prog1 (anexo) - Uso dos comandos fork() e exec()</u>

- ✓ Este exercício propõe o entendimento do uso das funções fork() e exec().
- ✓ Prepare o código de prog1.c, segundo os requisitos solicitados.
- ✓ Execute o programa e responda no relatório às questões colocadas no código do programa.
- ✓ Mostre no relatório o conteúdo da console durante a execução.

d. <u>Prog2 – Subprocessos Cooperativos</u>

- ✓ Construa um programa que simula a execução de um interpretador de comandos.
- ✓ Prepare o código do algoritmo apresentado em prog2.c segundo os requisitos solicitados.
- ✓ **Versão 1 do programa**: O comando não possui opções e argumentos
- ✓ **Versão 2 do programa**: Incluir a possibilidade de passar parâmetros e argumentos para o comando a ser executado, conforme sintaxe de comandos Unix:

  $> comando *opções argumentos*

  Opções: precedidas por '-' e seguindo uma letra

  Argumentos: nomes de arquivos

  Sugestão: Você pode utilizar a função *getopt()*.

8. Avaliação

A avaliação será feita nos dias 22 e 26 de novembro.

BOM TRABALHO

Anexo 1 – Exemplos de uso

(1)

```
main()
{
   int ret1, ret2;
   ret1 = fork();
   ret2 = fork();
   printf("Programa em execução.\n");
}
```

(2)

```
main()
{
   int ret;
   ret = fork();
   if (ret == 0)
      execl("/bin/ls","ls",0);
   else
      printf("Processo continua executando.\n");
}
```

(3)

```
main()
{
   int ret;
   ret = fork();
   if (ret == 0) {
      execl("/bin/ls","ls",0);
      printf("Quando este comando será executado ? \n");
      };
   printf("Por que a função printf anterior não foi executada?\n");
}
```

(4)

```
main()
{
   int ret;
   ret = fork();
   if (ret == 0) {
      execl("/bin/ll","ll",0);
      printf("Por que este comando foi executado ? \n");
      }
   else
      printf("Processo continua executando.\n");
}
```

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 encerrar
      } 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/ls", "ls", NULL);
            ***** O que acontece após este comando?
            ***** O que pode acontecer se o comando "execl" falhar?
      }
      exit(0);
}
```

prog2.c

```
Início
      Lê linha de comando;
      Enquanto não fim faça
            Início
                  Percorre a linha retirando o nome do comando;
                  Executa um fork para criar um novo processo;
                  Se processo filho então
                        Executa execl especificando o nome do comando  como
                        parâmetro;
                  Senão
                        Inicio
                        Executa wait para esperar que a execução do comando termine;
                        Se codigo retorno = zero então
                              Escreva "Executado com sucesso."
                        Senão
                              Escreva "Código de retorno = ", codigo_retorno;
                        Fim

                  Fim se;
                  Lê linha de comando;
            Fim;
Fim;
```