Universidade de Brasília
Departamento de Ciência da Computação
Disciplina de Programação Concorrente
Prova 1 Período: 2022/1 Data: 21/07/2022
Aluno: Matrícula:

Questões

1 (1.0 ponto) - Descreva as principais diferenças entre a programação sequêncial e a programação concorrente.

3 - Quando os processos cooperam, um processo pode afetar ou ser afetado por outros processos. Neste cenários, responda as questões abaixo.

a) (1.0 pontos) Defina condições de corrida.

b) (1.0 pontos) Apresente e explique um exemplo.

4 - Dada a solução abaixo para o problema dos leitores e escritores, responda as seguintes perguntas. Para cada pergunta, explique sua resposta apontando as linhas do código.

a) (1.0 pontos) É garantido que cada escritor possui acesso exclusivo ao banco de dados?

b) (1.0 pontos) É possível que um número qualquer de leitores acessem o banco de dados simultaneamente?

c) (1.0 pontos) Os escritores possuem preferência no acesso ao banco de dados?

d) (1.0 pontos) É livre de deadlock?

```
1  pthread_mutex_t turno, db;
2  int numLeitores = 0;
3
4  void * leitor (void *arg) {
5      while(1) {
6        pthread_mutex_lock(&turno);
7        numLeitores++;
8        if (numLeitores == 1) {
9          pthread_mutex_lock(&db);
10       }
11       pthread_mutex_unlock(&turno);
12       //ACESSA O BANCO DE DADOS PARA LER
13       pthread_mutex_lock(&turno);
14       numLeitores--;
15       if (numLeitores == 0) {
16         pthread_mutex_unlock(&db);
17       }
18       pthread_mutex_unlock(&turno);
19       //UTILIZA O DADO LIDO
20     }
21     pthread_exit(0);
22 }
23
24 void * escritor (void *arg) {
25     while(1) {
26       //PRODUZ DADOS
27       pthread_mutex_lock(&turno);
28       pthread_mutex_lock(&db);
29       //ESCREVE NO BANCO DE DADOS
30       pthread_mutex_unlock(&db);
31       pthread_mutex_unlock(&turno);
32     }
33     pthread_exit(0);
34 }
```

---

5 (3.0 pontos) - Resolva o seguinte problema: O estacionamento de uma universidade possui 30 vagas. Enquanto o mesmo possui vagas, não existe prioridade para quem estaciona. Quando o mesmo está cheio, forma-se uma "fila" (imaginária), onde professores possuem a mais alta prioridade, seguido pelos funcionários e após pelos alunos. Modele a entrada e saída do estacionamento para os "professores", "funcionários" e "alunos" (veja os protótipos das funções no verso da prova).

```
int capacidade = 30;
...

void main(argc, argv){
  ... //colocar a inicialização de variáveis
  criar_professores_funcionarios_alunos();

}

void * professores(void *arg){ //vários professores
       ... //colocar procedimentos para entrar no estacionamento
       sleep(50); //tempo estacionado
       ... //colocar procedimentos para sair do estacionamento
}
void * funcionarios(void *arg){ // vários funcionários
       ... //colocar procedimentos para entrar no estacionamento
       sleep(25); //tempo estacionado
       ... //colocar procedimentos para sair do estacionamento
}
void * alunos(void *arg){ // vários alunos
       ... //colocar procedimentos para entrar no estacionamento
       sleep(10); //tempo estacionado
       ... //colocar procedimentos para sair do estacionamento
}
```