

Primeira Atividade de Avaliação FSO 2020/1

Instruções para entrega

Os exercícios constantes neste documento se referem à primeira avaliação de FSO de 2020/1.

Os exercícios teóricos T1 e T2 devem ser respondidos em um arquivo de texto com extensão `md`, ex: `meunome.md`, e deve ser submetido no sistema MOJ no problema marcado como **Teóricos**.

A correção dos exercícios teóricos serão feitas de maneira assíncrona, ou seja, você não saberá se acertou quando enviar ao CD-MOJ.

O problema prático deverá ser submetido ao sistema MOJ no problema marcado como **Prático**. Este problema será corrigido pelo sistema e você, possivelmente, receberá a resposta adequada quanto a sua solução.

Caso o sistema de juiz automático fique fora do ar, a correção será feita de forma posterior sem mudança nos prazos de submissão.

Notas

A prova valerá até (sempre considerando o dia 28 de setembro de 2020):

- 100 pontos se todas submissões ocorrerem até 18h59
- 80 pontos se existir alguma submissão a partir de 19h00 até 19h59
- 70 pontos com submissões entre 20h00 e 20h59
- 60 pontos com submissões entre 21h00 e 21h59

Ou seja se o aluno fizer todas as submissões até às 18h59, a sua nota será no máximo 100 (dependerá ainda das correções das questões teóricas).

Se o aluno enviar submissões após 18h59 ele sofrerá penalidades, 20 pontos de penalidades até 19h59, 30 pontos se existir submissão até 20h59 e 40 pontos quando enviar até 21h59.

Para aluno PNE (sempre considerando o dia 28 de setembro de 2020):

- 100 pontos se todas submissões ocorrerem até 19h59
- 80 pontos se existir alguma submissão a partir de 20h00 até 20h59
- 70 pontos com submissões entre 21h00 e 21h59

Para gerar a penalidade será sempre considerada o horário da última submissão feita pelo aluno. A penalidade será a calculada sempre desta forma, mesmo que parte das soluções tenham sido enviadas dentro do intervalo de mais pontos. O valor de penalidade será sempre calculado tendo como base a última submissão.

[25 pontos] - T1: Tradição de família

Suponha que um processo está inconformado com ter de morrer após a recepção de um sinal do tipo SIGTERM. Resolveu, então, criar um filho para garantir a sobrevivência da família nestes casos. Com base no seu conhecimento sobre a função `fork()` e tratamento de sinais, explique por que no código abaixo o filho morrerá logo após o pai.

Indique uma alteração para garantir que o filho ficará bravamente esperando um sinal e terá outro filho que fará o mesmo se receber um SIGTERM e assim sucessivamente.

```
01: void trata_SIGTERM(int signum) {
02:     if (fork()) {
03:         printf("Meu filho continuara a tradicao da familia!\n");
04:         signal(SIGTERM, SIG_DFL); /* Reinstala tratador padrao e */
05:         raise(SIGTERM);           /* levanta SIGTERM para encerrar execucao*/
06:     }
07: }
08:
09: int main() {
10:     signal(SIGTERM, trata_SIGTERM); /* Instala o tratador de sinal */
11:     pause(); /* Interrompe a execucao e aguarda um sinal */
12:     return 0;
13: }
```

- Nota: verifique o manual da função `raise` invocando o comando `man raise`

[25 pontos] - T2: A respeito de `fork()` e `signal()`

Suponha que um processo invoca a função `fork()`. Imediatamente depois, este processo pai instala um tratador de sinais e vai dormir com o comando `pause()` (que interrompe a execução de um processo até que este receba um sinal). Quando começa a executar, o filho envia um sinal SIGALRM para acordar o pai. Quando recebe o sinal, o pai exibe uma mensagem e termina sua execução.

```
void trata_SIGALRM(int signum) {
    printf("Ai que sono! Queria dormir mais...\n");
}
int main() {
    if ((pid = fork()) != 0) {
        signal(SIGALRM, trata_SIGALRM); /* Instalacao do tratador de sinal */
        pause(); /* Pai espera ser acordado pelo filho */
    }
    else
        kill (getppid(), SIGALRM); /* Filho envia sinal para acordar o pai */
    return 0;
}
```

Descreva dois cenários em que este código apresentaria um comportamento diferente do descrito acima.