

Enunciado do trabalho-1 de Sistemas Operativos

Ano letivo 2021/22 - Universidade de Évora

Pretende-se implementar, usando a linguagem C, um simulador de Sistema Operativo considerando um modelo de 5 estados. O modelo deve incluir os estados READY, RUNNING, BLOCKED, e ainda os estados NEW e EXIT. Quando os processos estão em READY, e BLOCKED, estão em filas de espera do tipo FIFO (first in first out).

Os processos são definidos por um conjunto de pseudo-instruções-máquina representadas por números, que representam alternadamente:

- o tempo que o processo fica no CPU (estado RUNNING)
- o tempo que gasta no estado BLOCKED **depois de chegar à fila de BLOCKED**.

Por exemplo 2,3,5,3,1,3,1 significa que o processo (quando chegar a sua vez) fica.

--- 2 instantes no CPU (RUNNING); seguido de

--- 3 instantes no BLOCKED; seguido de

--- 5 instantes no CPU (RUNNING);

--- etc

Considere os processos definidos pelos seguintes pseudo-programas que iniciam no instante indicado em `programas[i][0]` (ou seja o instante inicial de P1 é 0, o instante inicial de P2 é 1 e o de P3 é 3,)

exemplo:

```
int programas[3][10] = {
    {0, 3, 1, 2, 2, 4, 0, 0, 0, 0 } ,
    {1, 4, 2, 4, 1, 1, 0, 0, 0, 0 } ,
    {3, 2, 1, 6, 1, 3, 1, 1, 0, 0 } };
```

Note que a partir de certa altura o programa tem sempre zeros (um ou mais zeros). Isso significa que após o último valor diferente de zero (o qual representa tempo de CPU) o processo passa para o estado EXIT e termina.

Implemente um simulador dos processos neste sistema tendo em consideração que:

- 1) Admite-se que a mudança entre estados é infinitamente rápida; depois os processos ficam nos diversos estados 1 instante de tempo, seguindo-se outra mudança de estados, e assim sucessivamente.
- 2) Os processos quando são criados passam para o estado NEW e permanecem nesse estado 1 instante de tempo. Note que mais do que um processo iniciar no mesmo instante, ficarão vários processos no estado NEW, durante um instante de tempo, e todos eles passarão para o estado READY no instante seguinte.
- 3) Os processos quando saem de NEW passam para o estado READY, respeitando a ordem pela qual entraram em NEW. No entanto, se a fila de READY estiver vazia, e o CPU (estado

RUNNING) não estiver ocupado um processo pode passar de NEW para READY e deste para RUNNING/CPU “instantaneamente”

4) Os instantes na fila BLOCKED, para cada processo contam assim que o processo chega à fila.

5) Os processos quando terminam, passam do CPU (estado RUNNING) para o EXIT, onde permanecem 1 instante de tempo.

6) Os processos depois de estarem 1 instante de tempo em EXIT desaparecem do sistema.

7) O escalonamento a implementar deve ser o algoritmo FCFS (first-in first-out) e Round-Robin com Quantum =3.

8) Os processos que saem de BLOCKED, passam para READY.

9) O número máximo de programas a dar entrada no sistema é de 10, e cada programa tem a dimensão máxima de 10 (1 instante inicial, mais 9 instruções alternadas de CPU e BLOCK).

10) Se no mesmo instante puderem entrar vários processos na fila de READY vindos de RUNNING, BLOCKED, e NEW, estes entrarão na fila de READY pela seguinte ordem: 1º os de BLOCKED; 2ª o de RUNNING/CPU; e por último os de NEW.

11) O programa deve ter como output, o estado de cada processo em cada instante.

Por exemplo:

instante | proc1 | proc2 | proc3

1	NEW		
2	RUN	NEW	
3	RUN	READY	
4	RUN	READY	NEW
5	BLCK	RUN	READY
6	READY	RUN	READY
7	READY	RUN	READY
8	READY	RUN	READY

etc.....

Este trabalho deve ser desenvolvido em grupos de 1, 2 ou 3 alunos até ao dia 12 de Abril, e submetido no moodle como um ficheiro .zip com os número de aluno no nome do ficheiro, ex “114444.zip” e deverá conter o código fonte do trabalho assim como um relatório em PDF. O relatório deve conter uma descrição sucinta da estrutura do programa proposto (máximo 4 pág) e o output dum conjunto de “programas” teste que será oportunamente disponibilizado no moodle.

Podem ser adicionadas a este enunciado notas e esclarecimentos adicionais de modo a clarificar alguma ambiguidade.