

Relatório Sistemas Operativos
Trabalho 2
2021/2022



Trabalho realizado por:
António Nanita nº 48407
Diogo Carreiro nº48729
Rodrigo Alves nº 48681

Introdução

Neste trabalho pretende-se implementar, usando a linguagem C, um simulador de um sistema operativo mais realista. Este simulador incluirá a simulação de uma memória linear, a execução de instruções com operações simples com variáveis, instruções de controlo, operações de input/output e uso de threads. Para isso iremos usar um modelo de 5 estados, com os estados NEW, READY, RUN, BLOCKED e EXIT, para os programas, e um modelo de 3 estados para as threads, que nos irão auxiliar na realização deste simulador. O objetivo do trabalho é aplicar a um determinado input o algoritmo de escalonamento Round Robin, com o quantum igual a 3. No nosso trabalho utilizámos a estrutura de dados Queue, para ajudar no escalonamento dos processos, sendo que foram criadas duas Queues, uma para guardar os processos que estavam no estado READY, e outra para guardar os processos no estado BLOCKED. Para além disso criámos duas estruturas chamadas TCB e PCB, que irá armazenar as informações relativas a cada processo ou thread, e que irá guardar as informações acerca da memória utilizada, respetivamente. O grupo decidiu que cada estado seria representado por um inteiro, de modo a facilitar tanto a leitura do código, como também na sua implementação. Neste trabalho foram nos sugeridas as estruturas mencionadas anteriormente pelo professor, sendo que foram adicionadas às mesmas variáveis complementares, que nos ajudaram na implementação do código apresentado.

Descrição do Programa

Para a representação dos processos/threads nos estados criámos as seguintes funções:

Print_New-Printa, para cada instante os processos que se encontram no estado NEW

Print_Ready-Printa, para cada instante os processos que se encontram no estado READY

Print_Run-Printa, para cada instante os processos que se encontram no estado RUN

Print_Blocked-Printa, para cada instante os processos que se encontram no estado BLOCKED

Print_Exit-Printa, para cada instante os processos que se encontram no estado EXIT

count_prog_and_thread-Função que retorna o numero de programa e de threads que o input contém

time_in_run_blocked-Função que irá atribuir para cada programa/thread o numero de tempos que cada um irá ser “corrido” no estado RUN, e o numero de tempos no estado BLOCK.

read_instructions_start- Função que define a primeira instrução de cada processo/thread.

read_instructions_end- Função que define a ultima instrução de cada processo/thread.

Remove_others- Esta função remove todas a threads que estejam ligadas a um dado programa que lhe foi atribuido o estado EXIT.

Execute_blocked- Esta função executa a instrução do processo no estado BLOCKED.

line_time_rr- Função que retorna o estado do simulador num dado instante.

All_exit- Função que verifica se todos os programas/threads estão no estado EXIT.

Para além das funções indicadas foram implementadas as seguintes funções sugeridas: Blocked2ready(), Run2exit_blocked_run(), New2ready(), Ready2run(), NewProcess(), RemoveProcess() e Execute() que ajudam na verificação de transições entre estados e na execução do algoritmo de escalonamento Round Robin.

Conclusão

Em suma, o nosso código apresenta algumas limitações, como por exemplo, quando é feita a leitura das instruções do ficheiro de texto, ao serem identificadas, algumas delas não são reconhecidas totalmente pelo programa, isto é, numa instrução do tipo JIFZ X, o valor de X não é guardado na variável que armazena o nosso input. Este problema limita-nos na execução da parte da memória, como também no número de instruções que cada programa/thread possua.