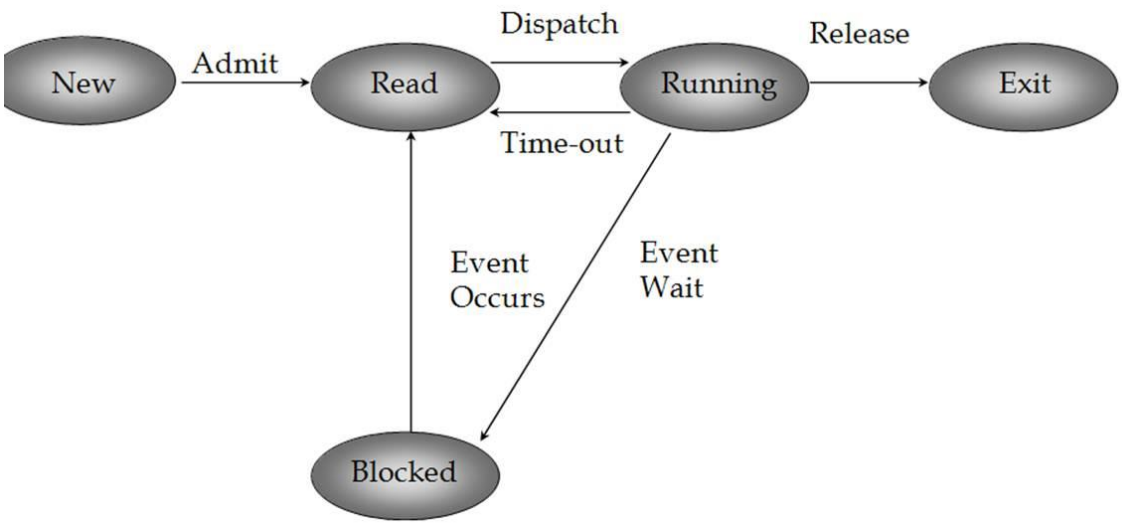
Universidade de Évora

Engenharia Informática

Sistemas Operativos I



Modelo de 5 Estados

Autores:

Raquel Gomes 31523

João Silva 32355

Docentes:

Professor Luís Rato

Professor Pedro Patinho

Índice

Introdução

Antes de mais, a realização deste trabalho encontra-se inserida na componente prática da disciplina de Sistemas Operativos I; a realização do mesmo foi-nos proposta pelo professor das aulas práticas, Pedro Patinho. Em relação à disciplna em si, encontra-se inserida na Licenciatura em Engenharia Informática da Universidade de Évora, mais propriamente no 4º semestre .

Falando do trabalho e daquilo que nos propomos a fazer, vamos fazer um pequeno resumos e, de seguida, especificar mais; generalizando: o nosso trabalho passa por, baseando-se no modelo de 5 estados, construir um "job scheduler" que tenha uma execução máxima de 10 processos em que cada processo tem um conjunto de instruções; este sistema, que será interativo e terá de usar o algorítmo "round robin", terá de ter um "command prompt" onde se pode lançar os programas e o "output" tem de ser guardado num ficheiro (e, neste pequeno resumo, englobámos toda a primeira parte); em relação à segunda parte, consiste em alterar a gestão de memória para um sistema de páginas.

Como podemos ver/ler a cima temos todos os passos que vamos ter de percorrer para que, no fim, tudo esteja a implementado e a fazer o que desejamos; apesar de não termos falado na implementação propriamente dita, achámos que não era necessário pois, essa mesma implementação, vai ser falada/construída à medida que este relatório ficar, também, próximo de estar finalizado.

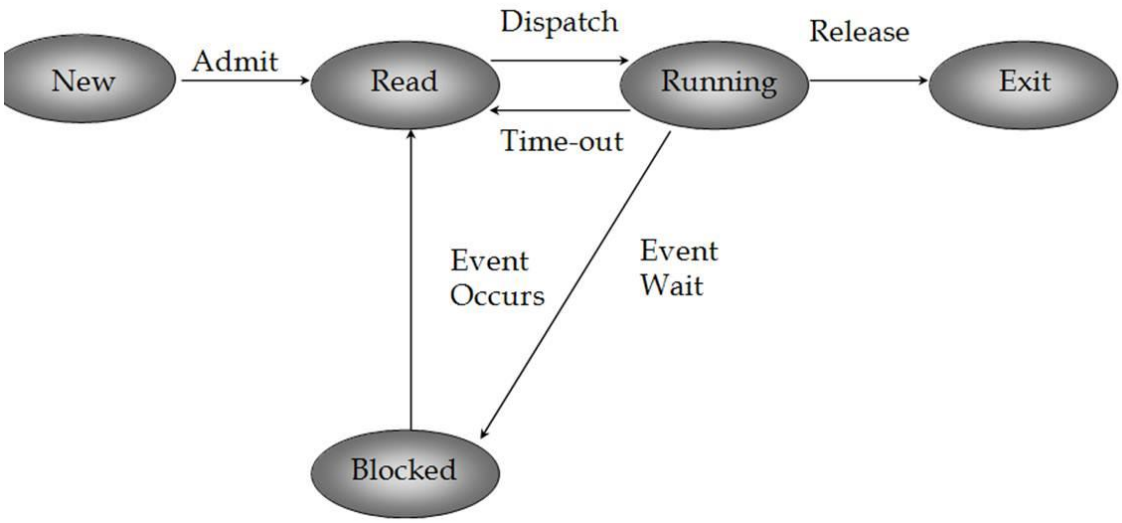
Depois de termos falado nos planos em si e de termos explicado, do nosso ponto de vista, o projecto que nos foi proposto, queríamos também salientar que nos encontramos motivados com o projecto e que sabemos que, cajo surja alguma dúvida, podemos contar com o docente da disciplina para nos esclarecer todas as nossas dúvidas.

Funcionamento

(Primeira Parte)

Nesta primeira parte vamos focar-nos, principalmente, no modelo de 5 estados, nas instruções, nas classes PCB e Escalonamento, no tratamento de instruções e no Escalonamento.

Modelo de 5 estados



No geral, neste modelo, temos que:

New: quando um processo é criado, ocupa a posição "new" ou "created"; nesta posição o processo espera a sua admissão no próximo estado ("ready"); esta admissão vai ser aprovado ou adiada, conforme outros processos que estejam, ou não, a correr;

Ready: quando um processo está nesta posição significa que está pronto a ser executado; no entanto, podem haver mais processos prontos a ser executados e, se se tratar de um uniprocessador, apenas um processo é exectuado de cada vez;

Running: um processo está nesta posição quando é escolhido para ser executado; este mesmo processo pode correr em dois modos: "kernel mode" ou "user mode";

Blocked: um processo pode ocupar esta poisção por vários motivos, no entanto, o mais provável é que esteja à espera que algum evento ocorra/se realize;

Exit: isto acontece quando um processo é terminado, ou seja, o processo ocupa esta posição quando termina ou quando é terminado pela execução do programa ("killed").

Instruções

Tal como foi dito, cada processo é constituído por um conjunto de instruções:

1- CPU- Vai executar um cálculo na CPU;
2- DISK- Vai executar um acesso ao disco (I/O);
3- NOP- Não executa nada;
4- GOTOBEGIN- Salta para o início do programa (PC <-- 0);
5- FORK- Duplica o processo e o novo processo vai para o início ("ready");
0- EXIT- Termina a execução do programa.

Classe PCB

Nesta classe, Process Control Block, o objectivo é fazer com o que PCB guarde os dados mais relevantes do programa; em suma, para dar a conhecer do seu funcionamento, passamos a explicar: vamos guardar o ID (isto para saber de que processo se trata), vamos o guardar o estado em que o processo se encontra (New, Ready, Running, Blocked, Exit), vamos guardar o tempo em que se inicia para saber o instante em que começa o novo processo; vamos ter um Array com todas as instruções que irão ser lidas no "txt"(o ficheiro de texto) e, por fim, vamos ter o PC (programm counter) para saber, num dado momento, qual a instrução que vai ser executada.

Classe Escalonamento

Nesta classe, vamos ter vários aspectos cruciais para o desenvolvimento do programa em si, tais como: uma variável que vai guardar os processos que foram lidos pelo ficheiro, uma lista que irá conter cada estado do modelo (já mencionados a cima), uma variável que serve para ver o tempo actual (tal como falámos na classe PCB), um "timeout" para impedir que o programa seja infinito, outra variável para que possamos saber qual o número exacto de processos que estão a ser executados num determinado momento da execução do programa, outra lista com todos os processos e o seu respectivo estado actual para que possamos ter o "output" que esperamos e, também, algo que nos informe se o disco está, ou não, a ser utilizado (para isto vamos utilizar um booleano).

Tratamento das instruções

Aqui vamos ter uma função "input" da classe acima e que vai receber o "txt" de que falámos; esse "txt" vai ter o formato: Instante Inicial (espaço) Instrução 1 (espaço) (etc). O objectivo era incluir também os estados mas algo correu mal nesse aspecto. No entanto, pensamos ser perceptível com tudo o que foi explicado e feito até agora.

Escalonamento

Aqui vamos poder ver a interação de alguns aspectos de falámos anteriormente, nomeadamente a relação do modelo de 5 estados e da sua respectiva explicação com as instruções que falámos mais à frente; em suma, o que vamos ter aqui vai ser: se o processo está "Blocked" vai para "Ready" por ter prioridade, caso um processo esteja a correr ele verifica se é a última instrução (0- EXIT) e, caso seja, ele sai; se não houver nenhum processo a correr e a próxima instrução for a terceira (3- NOP) quer dizer que não está a executar nada e que pode executar outro, se houver um processo a correr e a instrução for a primeira (1- CPU) ou a segunda (2- DISK) ele incrementa o Programm Counter; se a instrução for a quarta (4- GOTOBEGIN) ele coloca o Programm Counter a nulo; por último, caso seja a quinta (5- FORK) ele vai duplicar o processo e colocar o Programm Counter a nulo novamente; o objectivo é gerar o "output" de que falámos.

(Segunda Parte)

Nesta segunda parte não temos para apresentar tanto quanto gostaríamos de ter; queríamos, no entanto, dizer que, caso o tempo não fosse tão escassos conseguiríamos tornar esta parte mais simples do que parece tendo em conta que é basicamente matemática e definir se a página está a ser utilizada ou não. Posto isto, bastava criar algumas restrições. Obviamente que não nos vamos alargar muito mais que isto e, portanto, damos esta parte por concluída.

Conclusão

Em relação ao que foi feito até agora queríamos salientar que ficámos satisfeitos com o que fizemos. No geral, não conseguimos completar todos os pontos exigidos no enunciado do trabalho que nos foi proposto pelo professor; apesar disso, trabalhámos o melhor que conseguimos e tentámos a todo o custo cumprir o tempo que nos foi proposto.

Na primeira parte do trabalho o que conseguimos fazer foi positivo; trabalhámos bem com o modelo de 5 estados e, não só trabalhámos com ele, como também trabalhámos sobre ele o que demonstra a importância do que aprendemos não só nas práticas como nas teóricas; os aspectos negativos desta primeira parte foram algumas dificuldades nos "outputs" e com o algorítmo; no entanto, conseguimos ultrupassar essas dificuldades com a continuação do trabalho que era suposto fazer depois desses passos.

Na segunda parte do trabalho, foi-nos pedido que fizéssemos alterações de modo a trabalho com paginação de memória; no entanto, e infelizmente, essa matéria, para nós, está pouco cimentada e, dado o pouco tempo que nos resta, decidimos não usufruir da ajuda que nos foi proposta pelo docente Pedro Patinho.

Para concluir, achamos que todo o trabalho que foi feito durante o tempo que dedicámos ao projecto foi positivo; apesar de acharmos que podíamos ter feito mais (principalmente na segunda parte) achamos que o tempo também foi um pouco escasso uma vez que também tivémos outros trabalhos, alguns testes e, agora, exames; contundo, não queremos manchar o que não fizemos com desculpas relativas a aspectos que não se relacionam com a disciplina. No geral, aprendemos bastante com tudo o que fizémos, conseguimos cimentar bastantes conceitos e aproveitar para trabalhar a componente prática da disciplina.

Agradecemos, também, toda a ajuda que nos foi dada pelo docente.