



universidade
de aveiro

Jantar de Filósofos

Sistemas de Operação – Trabalho Prático de Grupo

Trabalho de Processos

Ana Cruz	nº 76351
Diogo Ferreira	nº 76504
João Maia	nº 76364

Trabalho de Threads

Eduardo Silva	nº 76354
Pedro Silva	nº 72645
Nuno Capela	nº 76385



Vida de um Filósofo

Um filósofo possui um ciclo de vida que consiste em ciclos de meditar, ficar com fome e comer.

A vida de um filósofo é iniciada a meditar. Após meditar, fica com fome e escolhe uma refeição. Quer a escolha recaia sobre esparguete ou pizza, primeiro irá buscar a comida. Se não houver doses disponíveis, efetua um pedido e coloca o seu número na fila de espera de pedidos do empregado, e espera até o empregado satisfazer o pedido e o acordar. Depois, irá novamente verificar se existem doses disponíveis.

Se existirem doses disponíveis, irá retirar uma para si. Se, depois de retirar a dose, verificar que não existem mais doses disponíveis, avisa o empregado, mas desta vez não fica à espera que ele satisfaça o pedido.

Após possuir a comida em sua posse, o filósofo irá buscar os talheres (dois garfos no caso de esparguete, um garfo e uma faca no caso de pizza). Se o filósofo desejar comer uma pizza, primeiro irá tentar ir buscar um garfo, e só depois uma faca. Caso não haja garfos ou facas, o filósofo efetuará um pedido ao empregado contendo o número de garfos e facas necessários, e esperará que ele o concretize, voltando depois ao início o processo de ir buscar ambos os talheres. Se o filósofo desejar comer esparguete, poderá ir buscar ao mesmo tempo dois garfos. Caso não haja garfos, o filósofo efetuará um pedido ao empregado e esperará que ele o concretize.

Após a operação, caso não haja facas limpas ou caso haja menos de dois garfos limpos, o filósofo avisará o empregado, mas não ficará à espera que o pedido seja satisfeito.

Após efetuar a sua refeição, o filósofo coloca os talheres na caixa dos talheres sujos, e volta a meditar.

Após um número definido de ciclos de vida, o filósofo morre.

Vida de um empregado

O empregado iniciará a sua vida no estado de repouso, ficando à espera de pedidos de filósofos. Quando algum filósofo o acordar, o empregado irá verificar se existem pedidos de pizza, de esparguete ou de talheres, por esta ordem. Por iteração, o empregado apenas irá satisfazer pedidos de um tipo.

Caso existam pedidos de pizza ou de esparguete, o empregado irá repor as doses da comida pedida e irá acordar o número de filósofos presentes na fila de espera que podem continuar a sua execução, ou seja, no máximo, o número de doses repostas (por exemplo, se forem respostas 10 doses, poderão ser acordados até 10 filósofos). Se existirem mais filósofos à espera que não sejam acordados, eles irão ser movidos na fila de espera, aumentando a sua prioridade.

Se o empregado estiver a responder a um pedido de talheres, irá lavar todos os talheres sujos. Se não existirem talheres sujos, irá esperar até que exista pelo menos um talher sujo. Depois de lavar os talheres, irá acordar o número máximo de filósofos que possam continuar a sua execução. Se existirem mais filósofos à espera, estes serão movidos na sua fila de espera, aumentando a sua prioridade.

Quando acabar de responder a um pedido, o empregado bloqueia se não existirem mais pedidos efetuados. Se existirem mais pedidos, o empregado irá satisfazê-los de seguida.

Quando todos os filósofos estiverem mortos, o empregado lava todos os



talheres sujos e morre.

Versão 2 (Apenas no trabalho com processos)

No trabalho de processos, existe outra implementação alternativa com algumas diferenças em relação à explicada anteriormente, que visa diminuir a atividade do empregado, mas consequentemente aumenta o tempo de espera dos filósofos.

Assim sendo, o empregado só irá responder a pedidos quando todos os filósofos estiverem bloqueados à sua espera.

Distribuição dos trabalhos no grupo

A solução descrita neste relatório é uma solução global. Para a implementação específica, o projeto foi dividido em duas implementações diferentes, seguindo a mesma solução. A implementação do projeto com processos, memória partilhada e semáforos foi efetuada por Diogo Ferreira, João Maia e Ana Cruz. A implementação do projeto com threads, monitores e variáveis de condição foi efetuada por Eduardo Silva, Nuno Capela e Pedro Silva.