

Relatório Trabalho 2

Sistemas Operacionais II

Émerson Três de Paula - 213996

Thiago Bell - 228527

Ambiente de Teste

Ubuntu Linux 16.04.1 LTS

Processador Intel i7-2600 4 x 3.40GHz (Suporte a Hyper Threading)

Memória 8 GiB DDR3

Compilador GCC 5.4.0

Semáforos

Essa solução representa cada garfo como um semáforo. Cada filósofo, representados por threads, devem tentar adquirir ambos os garfos adjacentes a eles para poder comer. Como um deadlock pode surgir devido a possibilidade da formação de um ciclo (cada filósofo pega o garfo da esquerda e ninguém consegue obter o da direita, por exemplo), deve-se evitar esse ciclo. Para isso, usou-se a solução sugerida em aula: fazer um filósofo pegar primeiro o garfo da direita e depois o da esquerda enquanto os outros fazem o contrário. Não haverá problema com postergação infinita, pois, o tempo que cada filósofo come (e ocupa os garfos) é limitado a um valor aleatoriamente entre 1 e 10 s. Isso permite que outro processo entre na fila de espera pelo garfo antes de o anterior faça o mesmo.

Variáveis de Condição

Essa solução utiliza variáveis de condição para fazer a sincronização dos filósofos. O ato de adquirir os garfos é feito ao mesmo tempo para a direita e para a esquerda, verificando se os filósofos vizinhos não estão comendo, isso evita um possível deadlock pois nunca será formado um ciclo. Para evitar postergação infinita, a solução empregada foi de fazer com que o filósofo que sai do estado '*eating*' imediatamente ao passar ao estado '*thinking*' faz o teste para acordar os filósofos vizinhos se eles estiverem no estado '*hungry*' esperando para comer.