Relatório EP2 - SO turma 94

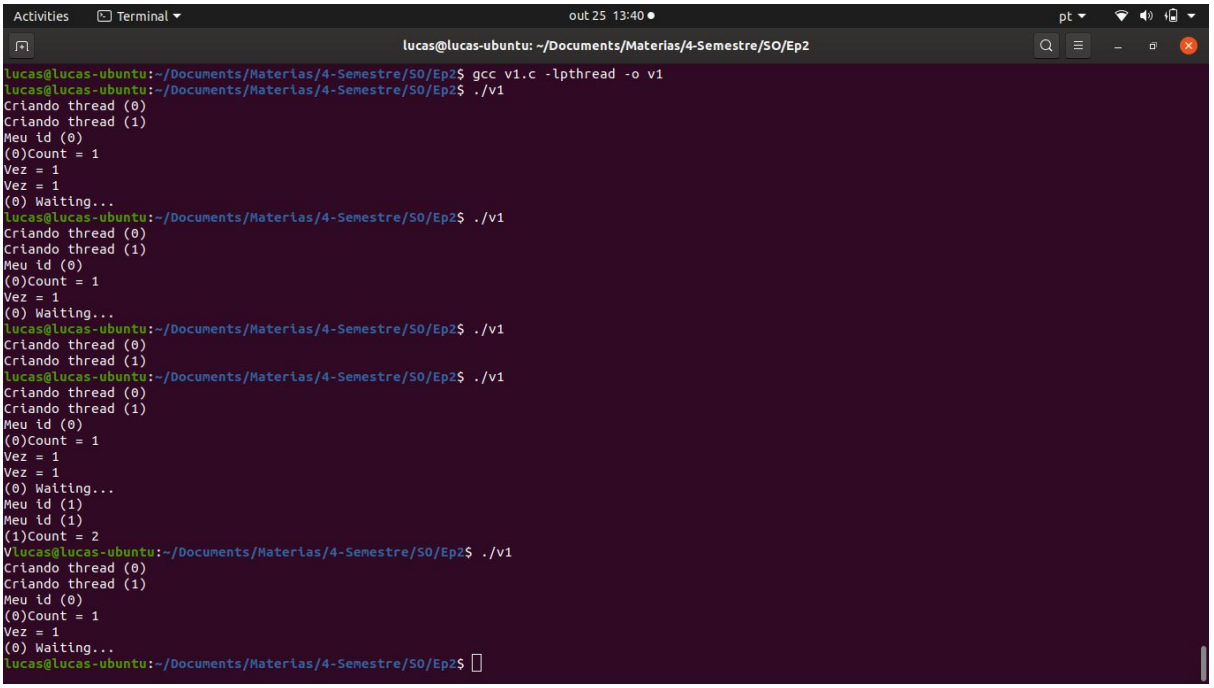
Lucas Mendes Sales , NUSP: 11270736

Utilizei como base o código do ep1, onde criava 10 threads com apenas um “Hello world”, reduzi para 2 o número de threads criadas e então busquei colocar dentro da função que passada para as threads um loop while similar ao presente no pseudocódigo fornecido na descrição da atividade, considerando como região crítica as alterações feitas nas variáveis globais.

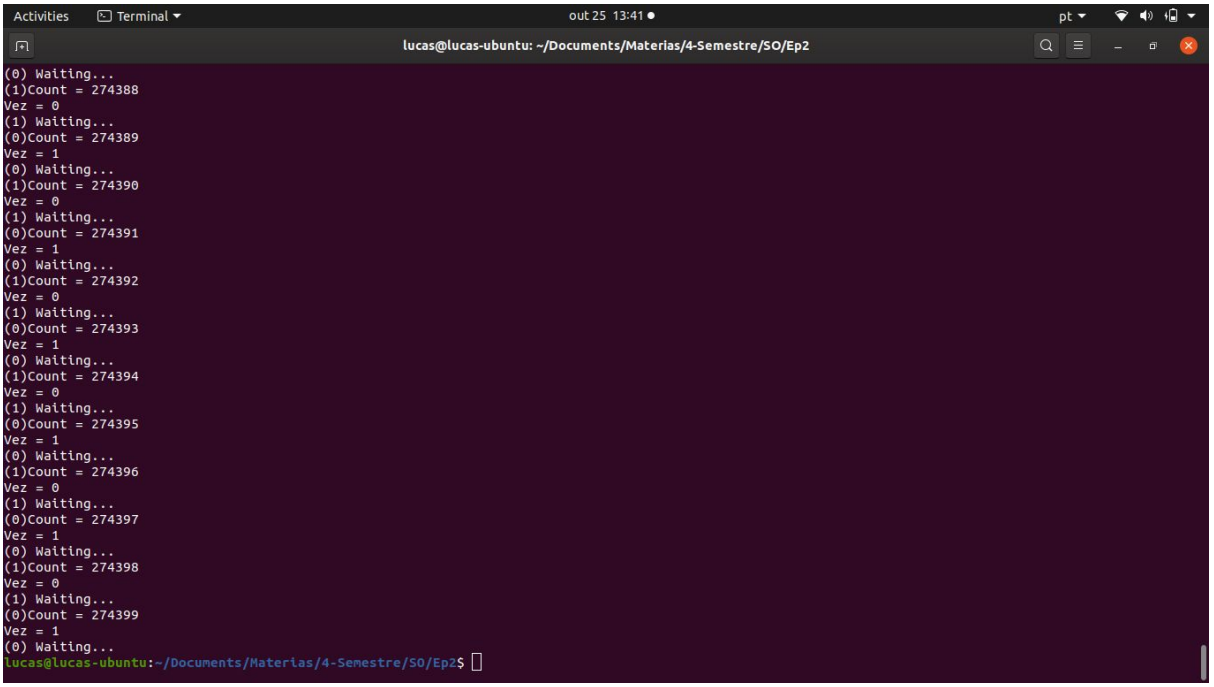
Foram criadas as variáveis globais *count, vez e running,* a variável running foi usada para controlar o tempo de execução do programa.

Durante as primeiras tentativas de execução, não obtive sucesso, pois não estava conseguindo passar o valor do identificador de maneira correta no momento da criação das threads, acredito que esse problema se deu por conta deste estar sendo passado para a função como o endereço da variável *i,* que era o iterador do loop *for* onde as threads eram criadas. Esse problema foi resolvido criando-se um vetor com valores fixos, e passando para a função o endereço do vetor na posição i.

Para que o programa não finalizasse sua execução antes de ser possível avaliar se a exclusão mútua estava ocorrendo de forma correta , foi criado um loop while que apenas incrementa uma variável n e dorme por 1s, quando n chega a 10, o valor de running muda para indicar o fim da execução.



Saídas antes do loop ao fim do programa.



Saída após o loop ao fim do programa.

Por fim, o uso da variável vez conseguiu garantir que as duas threads nunca entrassem na região crítica ao mesmo tempo, como esta é alterada ao fim do uso da região pela determinada thread, ela não será interrompida, e também fez que elas alternem tal acesso entre si, garantindo que não entrassem em um estado de eterna espera, e não é necessário levar em conta desempenho do processador para que isso aconteça.