

Laboratório 1 - Computação Concorrente - 2021.2

Lucas Guimarães Batista - 114136076

Atividade 1

Na minha máquina houve poucos casos em que a ordem de execução das threads foi alterada, mas esse caso ocorreu. Isso ocorre porque a partir do momento em que a thread é criada, nós não temos controle explícito sobre quando ela é executada, o que faz com que a ordem de execução seja basicamente aleatória.

Atividade 2

Além da diferença óbvia de cada thread ser identificada em seu print de “Hello world”, elas sempre foram executadas na mesma ordem, ou seja, incrementalmente conforme a ordem de criação.

Atividade 3

O programa aloca e preenche os argumentos para as threads e as cria em seguida, deixando-as prontas aguardando execução e partindo para a próxima alocação/criação. Dessa forma, podemos dizer que ele está funcionando como esperado.

Atividade 4

A única diferença notada entre a execução de *hello_join* e *hello_args* é que o print referente a finalização da thread principal é sempre feito por último quando executando *hello_join*, o que é exatamente o comportamento esperado graças ao uso da função *join()*.

Atividade 5

Código referente à implementação pedida está disponível [neste link](#).