

L4: The Relief of Vienna

It is September 1683. King Jan III Sobieski awaits the arrival of the main forces on Kahlenberg hill. Below lies a sea of Ottoman tents besieging Vienna. The situation is tense, but the King does not lose his confidence – he takes out his latest gaming laptop and decides to run a simulation of the attack before issuing the order for the real charge. Using modern technology, he intends to optimize the movement of the Hussar banners so that the strike is crushing. The fate of Christian Europe now rests in your hands.

Your task is to write a battle simulator in which the Hussars strike in three waves. You must ensure synchronization between units, logistics of weapons replenishment, and handling of sudden weather changes.

It is forbidden to use global or static variables.

Stages

Stage 1 (5 pts.)

The program accepts two arguments: N ($10 \leq N \leq 20$) denoting the number of banners and M ($2 \leq M \leq 8$) denoting the number of artillery threads. Start banner threads. Each of them must travel through a narrow gorge on Kahlenberg hill with a capacity of 3 units at a time (use a semaphore). The passage takes a random 80-120ms. Upon arrival, the thread prints: `CAVALRY <ID>: IN POSITION.`

Stage 2 (5 pts.)

The Hussars hold their attack until the enemy is weakened by artillery. Start artillery threads, which simulate shelling in a loop, reducing the shared value `enemy_hp` (initially 100) by random values in the range 1-6 every 400ms. The volley must be synchronized on a barrier. Every volley, one of the artillery threads prints: `ARTILLERY: ENEMY HP <enemy_hp>.` When the value of `enemy_hp` drops below 50, the threads stop firing and notify all waiting banners on a condition variable, which print: `CAVALRY <ID>: READY TO CHARGE.`

Stage 3 (7 pts.)

The battle consists of a total of three charges. In each of them, all banners must strike simultaneously. Use a barrier to synchronize the units. After it is reached by all N threads, only one of them prints: `CHARGE <NUM>!` (where NUM is the charge number: 1, 2, or 3). Then all threads sleep for 500ms (attack) and each of them decreases the enemy loss counter (`enemy_hp`) secured by a mutex by 1.

Between charges, the Hussars must retrieve new lances from the supply wagons. The number of weapon issuing stations is limited to 4 (use a semaphore). Retrieving a lance takes 100ms, during which the banner prints `CAVALRY <ID>: LANCE RESTOCKED.` After arming themselves, the banners reform the line on the same barrier before the next charge.

After performing all charges, the main thread prints: `BATTLE ENDED. ENEMY HEALTH: <enemy_hp>.` and if the value of `enemy_hp` is less than or equal to 0, it additionally prints `VENIMUS, VIDIMUS, DEUS VICIT!.`

Stage 4 (5 pts.)

Implement a dedicated thread for signal handling. The `SIGINT` signal safely interrupts the simulation and ends the program. The `SIGUSR1` signal simulates heavy rain – the thread handling signals prints: `RAIN AND MUD IS SLOWING DOWN CHARGE`. From this moment on, banners have a 10% chance to miss – they then print `CAVALRY <ID>: MISSED`. Remember to free all resources before ending the program.