How This Program Works: Commands and the Scheduler

This robot uses WPILib command-based programming. Commands describe actions like spinning the shooter. Subsystems group hardware like the shooter or intake. A scheduler runs every 20 ms and decides which commands should start, run, or end.

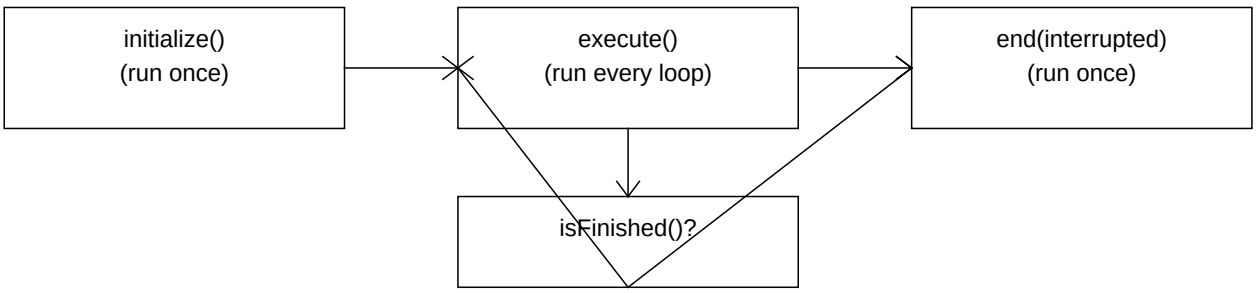
Where to look in our code

- RobotContainer.java: creates subsystems and connects buttons to commands.
- ShooterCommand.java: runs the shooter and logs counters on SmartDashboard.
- IntakeFalconCommand.java and IntakeSparkCommand.java: default commands for intake subsystems.

Command life cycle (what the scheduler calls)

Each command follows the same life cycle. The scheduler calls initialize() once, execute() every loop while scheduled, and end(interrupted) once when the command stops. isFinished() tells the scheduler when a command should end on its own.

Diagram 1: Command life cycle



Default vs non-default commands

A default command is always scheduled when nothing else is using that subsystem. In our code, the intake commands are default commands. They start once when enabled, run every loop, and only stop if another command needs that subsystem.

A non-default command runs only when scheduled by a trigger. Our shooter command is bound to the B button with whileTrue(). It starts when the button is pressed and is canceled when the button is released.

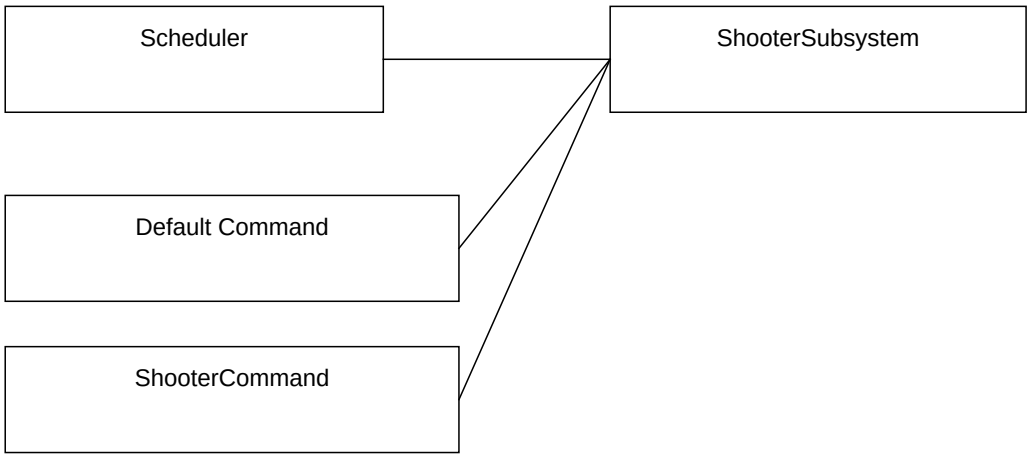
What the counters show and how we use this on the comp robot

What the dashboard counters show

- Default commands: InitializeCount is 1, ExecuteCount gets very large, EndCount stays low.
- Button command: InitializeCount and EndCount increase each press/release, ExecuteCount is smaller.
- cmdCancelled increases when the button is released because whileTrue() cancels the command.

Diagram 2: Scheduler, subsystems, and commands

Diagram 2: Scheduler picks which command owns a subsystem



How this scales to the competition robot

On the competition robot, we will use the same pattern. The drivebase will have a default command that reads the joysticks. The shooter, intake, and arm will each have button commands for actions like shoot, intake, and score. The scheduler guarantees that only one command uses a subsystem at a time and restores the default command afterward.

Key takeaways

- The scheduler is the traffic cop for commands.
- Default commands run in the background until something else needs the subsystem.
- Button commands run only while scheduled and often end by cancellation.
- The counters in ShooterCommand and Intake commands prove this behavior.