Backend Take Home

Agent Client Protocol is a communication protocol between Coding agents (Claude Code, Codex, Gemini, Cursor, etc) and clients.

claude-code-acp is an agent client protocol adapter for Claude Code developed by the team at Zed.

The task is simple. Implement a coding agent in Typescript that communicates with Claude Code over ACP, executable with bun, that has the following capabilities:

1. Send messages to Claude Code
2. Receive messages back from Claude Code
3. Approve and reject tool calls
4. Ability to set model, workspace directory
5. Create, edit and read files
6. Run shell commands

Extra credit features:

1. Streaming
2. Good Terminal UI
3. Resumable sessions.

You needn't use the ACP adapter, but you are also welcome to use it.

Spend less than 3-4 hours on this.

Expected Results

The following is acceptable output for the take-home test:

1. A working agent that can be used, and corresponding source code
2. Repository with well-labelled commits (not just a single commit at the end) showing the development process, including any mistakes.
3. If this wasn't possible in the allotted time, a write-up of the process, errors, your thoughts, issues encountered, etc can serve as a replacement.

What you will be judged on

This is a project that Southbridge team members have completed many times internally. You will be assessed on code quality and the process itself. Mistakes are okay!

AI Use policy

Southbridge is an AI deep-tech company that is also AI-enabled. Our internal team is *extremely* well versed in AI tools across the stack.

You are allowed to use AI assistance for solving this problem. As much as you want.

What is not allowed is *dishonesty*. All AI-use must be disclosed, explicitly mentioned, and flagged as such. You are responsible for any code handed in.

However, at the end of the day, what is being evaluated is your performance - either as a developer, a reviewer of code from an AI, or a software architect.

Ad Astra!