

Workstack

Tom Rochette <tom.rochette@coreteks.org>

December 21, 2025 — 77e1b28a

The workstack is a very simple idea I had while working. It is based on the concept of a [stack](#) as the name clearly implies. As you work, you, like a computer, process things one at a time and as new things need to be done, you either throw them in a todo list (a queue), or you start doing them right away (you stack them).

The workstack is a way to record notes about what you work on. As you work on something, you can either work on them to completion, or be interrupted by the necessity of working on another task. In the first case, tasks are simply written one after the other with their begin and end time. In the second case, items are also indented, such that it is possible to observe when a task forced you to “switch context”.

An example of this note taking format is as follow.

In this case, the person started working on tasks 1 and 2, then began working on task 3. As he began his work, he noticed that something else was necessary, which spawned task 4. While he was working on task 4, he observed something that could be done, but didn’t have to be done right away, which spawned task 5. As he completed task 4, he returned to task 3, but noticed that something else also had to be done, which effectively spawned task 6. During task 6, something else also interrupted him, which forced him to work on task 7. In this case, it could have been a coworker asking you for help on something. Task 5 could be a coworker asking for help as soon as you’re available, but not wanting to interrupt you.

Conceptually, you would want to always complete a stack of operations before moving to a new task. However, it is highly common while programming that a programmer will start going down such stack while working on code and then will not end up climbing back the stack, effectively not completing all he started working on.

This format thus allows a programmer (or anyone working on tasks that can spawn other tasks) to better track what they were doing and what they did and did not complete.

1 References

- [https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))
- https://en.wikipedia.org/wiki/Context_switch
- Parnin, C., & Rugaber, S. (2011). “Resumption strategies for interrupted programming tasks.” *Software Quality Journal*, 19(1), 5-34. <https://doi.org/10.1007/s11219-010-9104-9>