Still debugging with printf?

Speaker *Marc Khouzam*

Abstract text

With the advent of IoT and the widespread necessity for multi-core hardware, embedded applications are becoming exceedingly difficult to debug. Estimates put the time spent debugging at above 50%. Better solutions are essential for success in this context.

Recent advances in debugging technologies improve productivity and reduce lead time. Dynamic-printf, which allows inserting printf-equivalent instructions at run-time, remove the need for the all-too-common compile-deploy-debug-repeat cycle; reversible debugging lets developers undo execution and literally step backwards as well as forwards; non-stop debugging enables users to only interrupt and examine subsets of threads.

A key factor that enabled these advancements is open source; it has allowed open collaboration to create the best possible tools. This presentation covers many such features using open source tools like the GNU Debugger (GDB) and the C/C++ Development Toolkit (CDT). Topics covered include:

- Dynamic-printf
- Reversible debugging
- Non-stop debugging
- Global breakpoints
- Dynamic tracepoints
- Operating System awareness
- and more