

Appendix G – GSOC

GSOC project ideas



Difficulty on a scale 1–5

Generally all of our projects require C programming and prior experience with C is a prerequisite for all of them. Prior experience with Plan 9, while not technically a prerequisite, is extremely helpful and any student is advised to at least carefully read the Plan 9 documentation before applying.

MPEG-4 decoder (4)

The goal of this project is to create a video decoder to allow native playback of MPEG-4 video. This is a very challenging project that requires a very solid background in C. Potential mentors: cinap_lenrek Skills: Reading and understanding the MPEG-4 documentation which probably requires a certain minimum of mathematical skills, writing and debugging complex programs

Sam scrolling and other GUI work (2)

Scrolling in the text editor sam is currently very erratic. The goal of this project is to improve on this situation. This project does not require much programming per se but it does require analyzing, understanding and improving existing code. This project would probably be extended to fixing other GUI problems. Potential mentors: cinap_lenrek, Ori_B Skills: being able to innovate, coming up with algorithms for GUI problems such as scrolling

Improved APE (2)

Our ANSI/POSIX emulation layer is currently very unpolished and improving the coverage would be a worthwhile project. This is not very difficult but probably a lot of work. Potential mentors: cinap_lenrek, Ori_B Skills: reading ANSI/POSIX documentation, debugging code written by other people

Drivers (3)

This project would consist of multiple drivers (to be decided) that need to be written. This project is modestly challenging. Potential mentors: cinap_lenrek Skills: general systems programming, prior OS development experience helpful

DjVu/CHM/Jbig2 support (2)

Our document reader page(1) needs DjVu, CHM and Jbig2 (PDF encoding) support. Another project that's not particularly challenging but a lot of work. Potential mentors: cinap_lenrek, Ori_B Skills: reading and implementing documentation for various document standards, some experience with image processing probably helpful

Improving acid (3)

Our debugger acid(1) lacks several useful debugging features that might be worth looking into, such as watchpoints. The goal would be to research what might be good features and implement them. This may require kernel changes. Potential mentors: cinap_lenrek Skills: general systems programming, familiarity with low-level concepts

Implement ECDSA and ECDHE for TLS (2)

This would be implementing the ECC cipher suits for TLS (rfc4492) with libsecs ec(2) functions. This should be a relatively easy task for beginners. Potential mentors: cinap_lenrek Skills: reading standards, reading code, cryptography

Version control hosting support for werc (1)

Werc is a web anti-framework written in rc, the Plan 9 shell. rc-httdp is an HTTP server written in the same language. 9front uses git for version control -- but nobody's done much work on making 9front able to *host* git repos. This project would involve setting up werc and rc-httdp on a 9front System, then teaching werc to work with hg to host repositories. For bonus points, test your changes on a UNIX system to make sure it's portable! Potential mentors: khm Skills: web stuff, mercurial, rc scripting **Note:** 9front has recently switched from mercurial hosted on 9front to git hosted on 9front, so this project may now be obsolete. (Pending redaction.)

Internet of things project: use 9p for sensor networks (1–2)

Write a 9p server and client (as you see fit) on the orchestrator and nodes. Create and evaluate text-based protocol for 9p-based sensor network orchestration. You can use rpi for everything, but are also free to choose a low-power solution for the nodes if you're up for some low-level C programming experience. Potential mentors: jpm Skills: soldering, theory of sensor networks, C