Nick Black <`nickblack@linux.com`> **http://dank.qemfd.net**
855 Peachtree St NE, Atlanta, GA 30308 US Citizen (b. 1980–10–05)

We work in the dark. We give what we have. We do what we can.
Our doubt is our passion, and our passion is our task. The rest is the madness of art.
–Henry James, *The Middle Years*

Objectives

To continue researching, designing, implementing, deploying and maintaining innovative solutions for keeping the bits flowing and cachelines humming. To snatch order from a teeming chaos's complexity. To see the fruits of my labors entrusted with the packetized dreams of millions. To gaze back upon fifty-hour debugging sessions, seeing past the gnashed teeth, utter fatigue, the too-long compiles, and *know* the problem was solved. To—with a little help from some good code—make the world a better place.

I am actively seeking full-time employment in high-performance/scientific computing, systems programming, compiler design, or security.

Open Source

See http://dank.qemfd.net/dankwiki/index.php/Hackery for more details.

I passionately believe in the many societal benefits of open source, and actively participate in the Free Software community.

- Author and maintainer of numerous open source packages. Examples include:
  - *libtorque*, a multithreaded, architecture-adaptive event library,
  - *cubar* and *libcudest*, reverse-engineering tools for NVIDIA's CUDA, and
  - *ptracer*, an instruction trace generator.
- Contributed accepted patches to the Linux kernel (Matrox driver, IPv4 stack, and hugetlb filesystem), Wireshark (IPv4 analysis), DynamoRIO, OProfile, strace, OpenSSL, APT-secure, iproute2, bridge-utils, x86info, Snort, and other projects.
- Maintained IBM's NGPT (Next Generation POSIX Threading) kernel patchset.

Employment and Education

- **2011/01–Present: Senior Compiler Engineer, NVIDIA (Austin, TX)**
  - General development on NVIDIA's toolchains for Fermi and Kepler GPUs and Denver CPUs.
- **2010/08–Present: Georgia Institute of Technology, Doctor of Philosophy in Computer Science** ***On indefinite hiatus.***
  - Advised by Professor Tom Conte as a member of the `comparch` and TINKER research groups.
  - Investigated architecture, operating system, language and compiler support for high-performance network servers.

- **2010/05–2010/09: Chief Engineer, AccelerEyes LLC (Atlanta, GA)**
  - Led design work for the libJacket™ CUDA primitives library. Assisted ongoing design of the Jacket™ MatLab® plugin.
  - Implemented parallel GPU logical and numeric codes, including multidimensional convolutions, decompositions, sorts and searches.
  - Redesigned CPU-GPU memory transfer to use sharing, pinning and zero-copy, all atop a unified Bonwick-Alexandrescu allocator.
  - Reimplemented build system, autotesting, and office network from scratch.

- **2008/08–2010/05: Georgia Institute of Technology, Master of Science in Computer Science**
  - Research project: "Epicycles, Flywheels, and (Widening) Gyres: UNIX I/O Slouches Toward Multicore NUMA" (2009). Explored the state of the art in large-scale, massively parallel servers, and how current APIs ought change. Results include the open *libtorque* library and a 2010 paper, "Portable Multithreaded Continuations for Scalable Event-Driven Programs".
  - Sixteen classes, with twelve from among Systems and Information Security.
  - Teaching Assistant, "High Performance Computer Architecture" (CS6290—Superscalar, OOO, and manycore microarchitecture).

- **2005/12–2009/11: Principal Engineer, McAfee (Alpharetta, GA)** ***Impact Engineer of the Year, 2007–2008.***
  - Sole backend developer for the Secure Web Proxy Service, a managed Web security/control system.
  - Worked with academia in anti-bot efforts, intimately studying and engaging the Storm botnet. Developed the *argus panoptes* distributed anti-bot platform, and with it battled botnets across the IPv4 address space.
  - Lead developer of the IronNet™ appliance, making use of techniques including latent semantic analysis, clustering, and Markovian discrimination to prevent data leakage. This included the *snare* ICAP server.
  - Assisted development of image spam detection in IronMail™ and reputation protection in PhishRegistry™.
  - Developed *garuda*, a CheckPoint® OPSEC module interfacing with the TrustedSource™ reputation service.
  - Developed the *HURLBAT* protocol testing tool, and SMTP, HTTP and ICAP modules.
  - Oversaw development of hardware, Knoppix®-based bootable media, and custom tools for competitive field operations.
  - Rewrote the build system for IronMail™ and related products from scratch. Code review, automated building, virtualized testing, and unit testing were instituted along the way.
  - PCT/US2008/051869. Detecting Image Spam. 2008-07-31. **US Patent**

- **2000/08–2005/12: Senior Software Engineer, Reflex Security (Atlanta, GA)** ***Co-founder.***
  - Led research team, focused on intrusion prevention, parallelized and distributed intrusion detection, and multiple pattern matching. Implemented several techniques later published by academia. Led interviewing.
  - Sole developer of code for the Reflex Interceptor (now Reflex IPS™ and Reflex MG™), a Layer-2 bridging NIPS running Linux. This included:
    - *tako*, an IPS application statefully analyzing multiple GigE links inline and in real time, performing a forwarding verdict on each frame via use of `mmap(2)`ed packet sockets + custom `netlink(7linux)` messages, and
    - *geso*, an SMTP proxy making use of a Kaspersky™ Anti-Virus backend to filter mail inline and in real time, designed to be trivially extended to other store-and-forward protocols.
  - Assisted development of build systems, automated testing and benchmarking tools, and backend platform configuration management.
  - Assisted development of kernel patches to expand the netlink socket infrastructure and filter on `mmap(2)`-backed sockets.
  - PCT/US2004/023739. System and Method for Threat Detection and Response. 2005-02-03. **US Patent**

- **1998/12–1999/12: Teaching Assistant, GT College of Computing (Atlanta, GA)**
  - Recitations with ˜15 students, plus one-on-one meetings and grading of exams and homeworks. Classes included:
    - "Programming Language Principles" (CS 3411—Language design and comparative programming linguistics).
    - "Models and Translation" (CS 2330—Parsing, interpretation and compilation, virtual machines).
    - "Control and Concurrency" (CS 2430—Parallel computing, UNIX systems programming and ANSI/ISO C).
    - "Instruction Set Architecture" (CS 2760—Assembly language, stored programs, and architectural models).

- **1998/09–2000/02, 2004/01–2005/05: Georgia Institute of Technology, Bachelor of Science in Computer Science**
  - Specializations in Theory, Systems, and Networking.
  - 2000 ACM Programming Team, GT Team A ("Gold") (Captain).
  - 1999 ACM Programming Team, GT Team B ("White") (Captain).
  - Research under Professor Wenke Lee (Topics: IDS, botnet models, anti-bot tech).

Awards and Distinctions

- Credited corrections to Donald Knuth's *The Art of Computer Programming* and George Varghese's *Networking Algorithmics*.
- Impact Engineer of the Year, 2007-2008 (McAfee / Secure Computing).
- Faculty Honors, Georgia Tech 1999.
- Dean's List, Georgia Tech 1998.
- Third place, 1998 Questions Unlimited! National Academic Bowl Championships (New Orleans, LA).
- Second place, 1997 Questions Unlimited! National Academic Bowl Championships (Chicago, IL).

Skills

Algorithmic thinking. ANSI/ISO C. Development in the UNIX environment. Design and analysis of algorithms. Network programming. Effective use and internals of Linux, FreeBSD, their associated libc's and threading implementations, GCC, GNU binutils, LLVM and the open source toolchain. Robust system design. Network security. Intrusion prevention. Establishment and detection of covert channels. String algorithms. Applications of automata theory. Parallel algorithms and design for multicore/manycore machines. Design of programming languages. POSIX APIs along with Linux and FreeBSD extensions thereof. Analysis of binaries. Most major network protocols and their primary open source implementations. Effective use of cryptography. High-throughput, low-latency I/O models, scalable I/O and its implementation, zero-copy networking, and network hardware design. Tools of network enumeration and domination. x86 assembly language, including MMX and SSE. Computer architecture. Compiler design. Elegant system administration. Bayesian methods. Combinatorics, stochastics, topology, analysis, *ad nauseam*.

- **Operating Systems:** Linux (12 years), FreeBSD (6 years), Solaris (3 years), Windows NT (3 years)
- **Languages (expert—minimum 10 active years):** C, Bourne shell, x86 and MIPS assembly, Prolog, Lisp, XSLT/XPath, GNU Make
- **Languages (professional—minimum 5 active years):** C++, Python, Erlang, Haskell
- **Languages (amateur—minimum 1 serious project):** SPARC, PTX, JVM and m68k assembly, Java, ML, Scheme, JavaScript (and XUL), Scala, Clojure
- **Technologies:** GCC, LLVM, Pthreads, OpenMP, CUDA, OpenCL, Berkeley sockets, ACE, Flex/Bison, ANTLR, LaTeX, Apache, Postfix, MediaWiki, DJBDNS, nginx, Bugzilla, VPNs, SSH, DocBook, GDB, valgrind, strace, IDAPro, Gecko/XULRunner, nmap, OProfile, perf, tcpdump, Pin, DynamoRIO, GnuTLS, OpenSSL, Avahi, iptables, Git, svn, Portsnap, ipfw, LARTC, socat, KVM/QEMU, Xen, VMWare ESXi, BLAS, ATLAS… If it can be done on a Linux box, I've probably done it.

Last updated April 9, 2011. The most recent version can be found on my wiki. LaTeX source is available at GitHub.