Porting UNIX to Windows NT

David Korn
AT&T Labs
dgk@research.att.com

January 8, 1997

MOTIVATION

- Large collection of Windows software
- More familiar GUI
- Lots of existing UNIX applications
- Management requires Windows NT
- UNIX is simpler
- Enable UNIX development for Windows
- Bridge gap between UNIX and Windows

GOALS

- Minimal change to existing UNIX source
- Minimal performance overhead
- Ability to mix and match UNIX and Windows
- Windows NT and Windows 95
- Readily available – including source
- Learn about Windows NT and Windows 95

FEATURES OF WINDOWS NT

- Mach-like Micro Kernel
- Threads
- Modular Design
- Virtual Memory Model
- Level C2 Security
- Win32 API
- Runs DOS Applications
- POSIX and OS/2 Subsystems
- Networking Support
- UNICODE

ADVANTAGES OF WINDOWS NT

- Low Cost
- Lots of Software
- Controlled by Microsoft
- Source Portability
- Similar to Windows 95
- Many Experts

DRAWBACKS TO WINDOWS/NT

- Complexity - >1000 calls
- Subsystems are separate
- Controlled by Microsoft
- Not very innovative
- Too much like VMS
- Not many experts

PROBLEMS

- Case sensitivity + namespace
- Pathname syntax
- `<nl>` vs. `<cr><nl>`
- Signal handling
- Incomplete interface
- Permission mappings
- Building shared libraries harder

PORTING CHOICES

- Rewrite applications
- Use middleware framework
- Use POSIX subsystem
- Microsoft C library
- Use `NuTCracker` library interface
- Use `Portage` library interface
- Use `OpenNT` when available
- Write UNIX/POSIX library interface

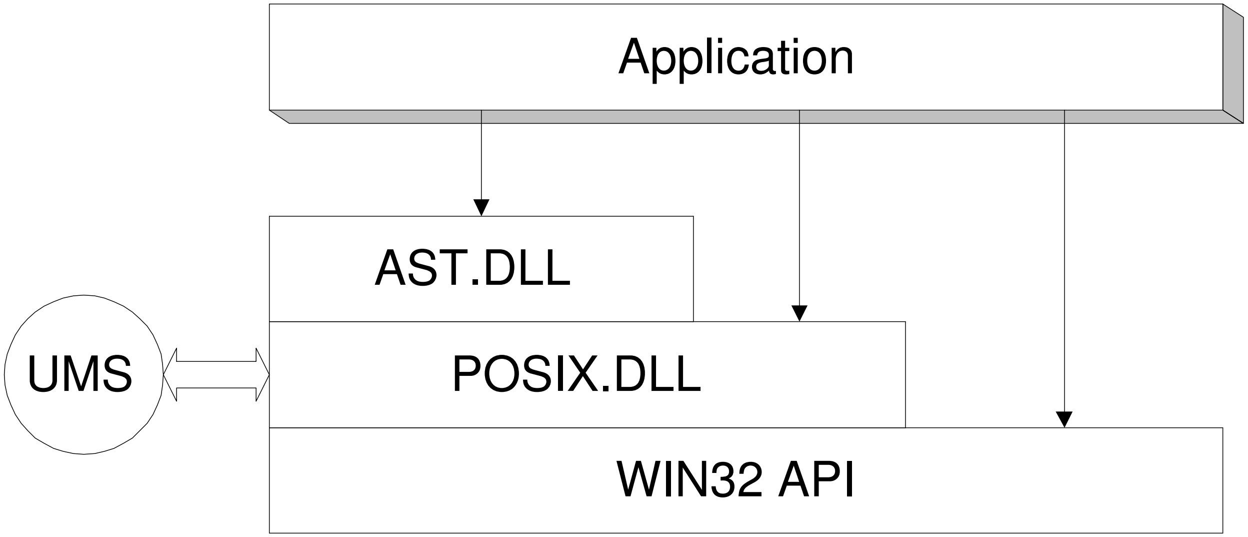
THE POSIX SUBSYSTEM

- Quick way port some UNIX application
- Can’t call NT application from POSIX
- No debugger support
- No support for tape or network naming
- Not supported on Windows 95

OUR POSIX LIBRARY - UWIN

- Written using WIN32 API calls
- Virtually all UNIX system calls
- `stdio` library based on `sfio`
- Microsoft C library for remainder
- Supports hard links/symbolic links
- Memory mapping and dynamic linking
- System V shared memory, semaphores, fifos
- BSD sockets over Winsock
- Setuid/setgid programs on Windows NT

UWIN Architecture



Application

AST.DLL

UMS

POSIX.DLL

WIN32 API

UWIN Server Process for NT

- Starts at boot time
- Run as Administrator – `uid` 0
- Creates `/etc/passwd`, `/etc/group`
- Creates security tokens
- Runs `/etc/rc`

CURRENT STATUS

- Version 1.14 4Q96
- C compiler wrapper for Visual C/C++
- *Practical Reusable UNIX Software* Tools
- Most X/Open interfaces
- Undergoing X/Open conformance testing
- `libast`, curses, X11R6 and other libs
- About 160 UNIX tools
- `ksh93` with editing and job control
- AT&T `nmake`
- New BSD `vi`
- telnet and rsh daemons

INNOVATIONS

- Windows shortcuts appear as symlinks
- `/dev/proc` and `/dev/fd`
- `/dev/windows` for windows events
- `/dev/clipboard` for clipboard
- Copy/paste for console windows

PERFORMANCE TESTS

- `sfio` benchmarks
- Andrew benchmarks
- Some `lmbench` benchmarks
- Homegrown benchmarks

		UWIN		WIN32		LINUX	
test	size	Kb/s	ratio	Kb/s	ratio	Kb/s	ratio
fwrite	10000K	15923	1.27	20408	1.00	11709	1.72
fread	10000K	36231	1.02	36764	1.00	9671	3.76
revrd	10000K	32679	.94	30303	1.00	19960	1.51
fw757	10000K	13227	.81	10593	1.00	9416	1.13
fr757	10000K	24154	1.13	27472	1.00	9199	3.02
rev757	10000K	10989	.67	7731	1.00	15128	.49
copy&rw	10000K	9940	1.01	9999	1.00	7173	1.39
seek+rw	2000S	34566	1.12	38672	1.00	8974	2.14
putc	5000K	5720	.87	5020	1.00	2393	2.09
getc	5000K	10245	.98	9960	1.00	3194	3.14
fputs	50000L	97656	.86	84459	1.00	57102	1.49
fgets	50000L	126262	.61	77399	1.00	93283	.83
revgets	50000L	72254	.29	21132	1.00	67114	.32
fprintf	50000L	8567	.80	6838	1.00	8968	.76
fscanf	50000L	10888	.93	10154	1.00	8933	1.14

TABLE 1. Stdio timings

		UWIN		WIN32		LINUX	
test	count	#/s	ratio	#/s	ratio	#/s	ratio
open/close	10000	3472	1.52	5291	1.00	22222	.24
create/delete	10000	233	3.36	783	1.00	3215	.24
readdir-2	1000	1040	2.26	2457	1.00	3571	.68
readdir-500	1000	236	1.23	291	1.00	221	1.31
system	100	7	2.31	17	1.00	35	.48

TABLE 2. Syscall timings

FUTURE

- Case sensitive file names
- Mount table
- Rest of inet daemons
- `tksh`
- Registry file system
- X/OPEN Conformance testing
- I18N based on UFT8
- 64-bit file support
- *n*-DFS
- Multi-format Documentation
- Run SCO/Linux binaries