

Porting PostgreSQL To The Win32 Platform

BRUCE MOMJIAN,
SOFTWARE RESEARCH ASSOCIATES

August, 2005



Abstract

This presentation outlines the challenges of porting PostgreSQL from Unix to the Win32 platform.

Pre-8.0



Problems With Cygwin

- Installation requirements
- Installation bloat, security concerns
- Fifty-connection limit
- Performance

Constraints

- Do not break the Unix build
- Do not make the code harder to understand
- Limit code duplication for the new port
- Minimize the amount of new code for the port

Solution — Teamwork

Commercial Ports

- PeerDirect
- SRA
- Connix

Target Operating System

- Windows NT, 2000/2003, XP
- NTFS

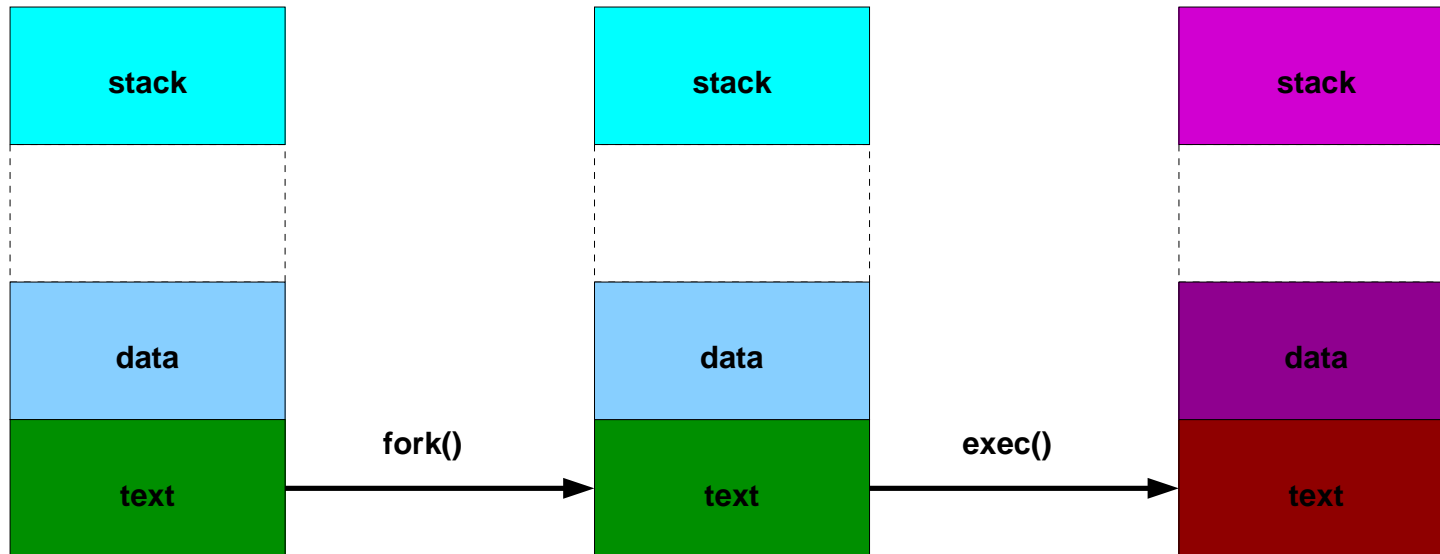
Toolkit - MinGW

- libraries map Unix calls to Win32 calls (MinGW)
- provides Unix build environment (Msys)

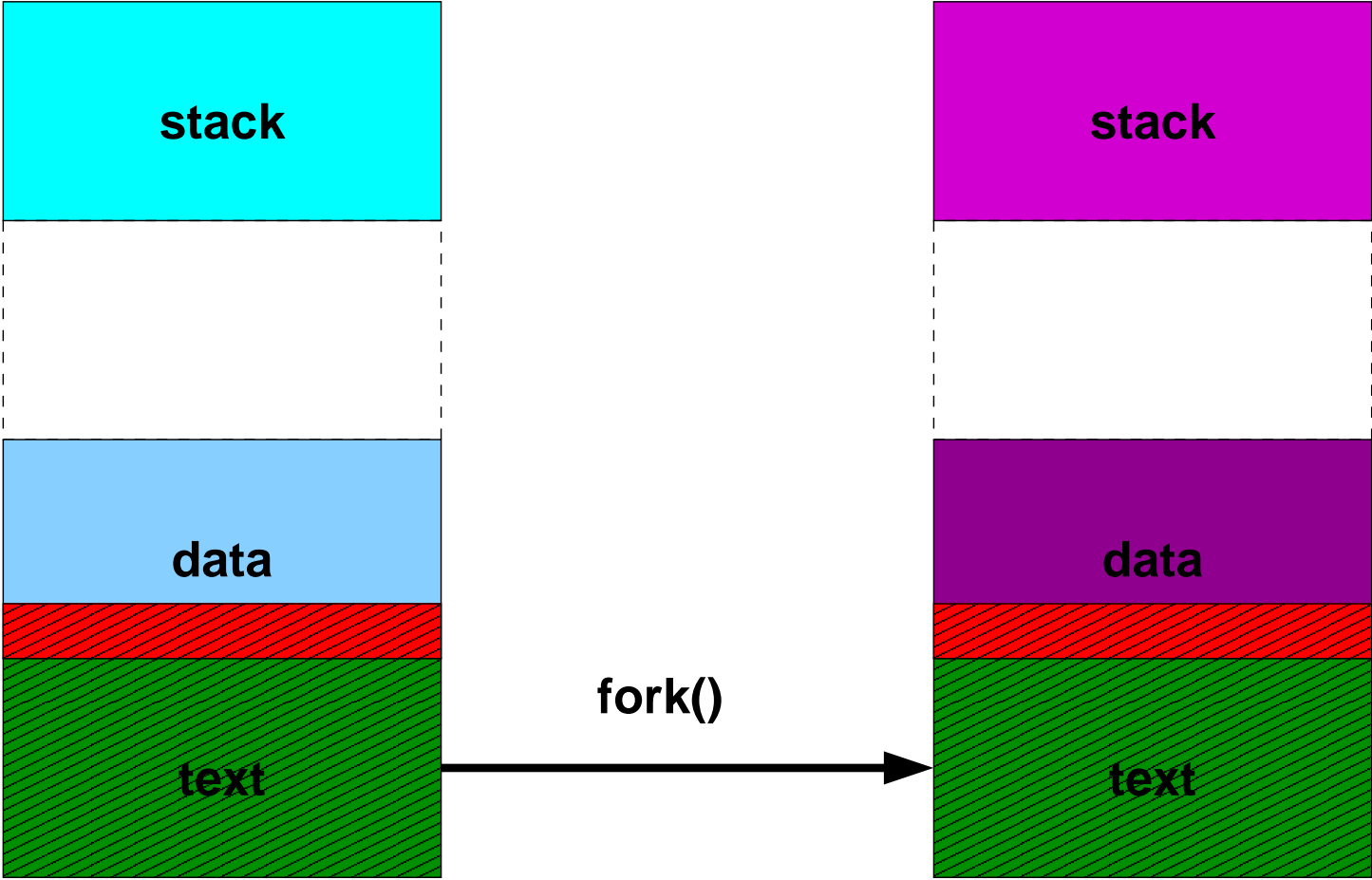
Major Issue - Fork vs. CreateProcess

- passing variables inherited via fork
- attaching to shared memory that is usually inherited

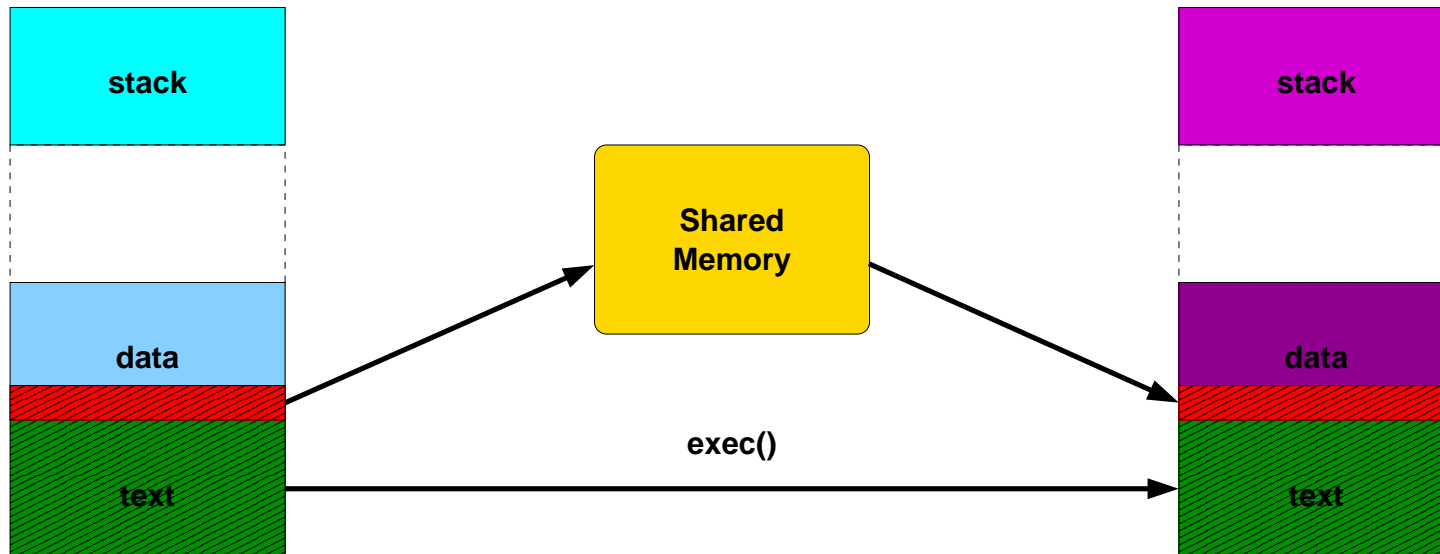
Typical New Unix Process



New PostgreSQL Process on Unix



New PostgreSQL Process on Win32



File System

- '/' vs '\'
- root directory, / vs. C:\
- carriage return / line feed
 - COPY
 - open()
 - character checks
- .EXE extension
- symlinks vs junction points
- file system sync() vs per-file commit()

File System Atomicity, Concurrency

- rename() and unlink() fixed using open() flags
 - FILE_SHARE_READ
 - FILE_SHARE_WRITE
 - FILE_SHARE_DELETE

Command Line

- `system()`
- `'&'` vs `START /B`
- `cp/copy`, `rm/del`
- `copydir` as service
- single vs double quotes

Environment Variables

- \$HOME directory
- \$SHELL vs \$COMSPEC
- \$EDITOR defaults

Shell Scripts Convert to C Programs

- `initdb`
- `pg_ctl`
- `create*`, `drop*`, `vacuumdb`, `pg_dumpall`

Interprocess Communication

- signals, thread listening on named pipe
- cancel key handling
- `popen()` when run from a service

/port Centralization

copydir.c	getrusage.c	pgsleep.c	srandom.c
crypt.c	gettimeofday.c	pgstrcasecmp.c	strdup.c
dirmod.c	inet_aton.c	pipe.c	strerror.c
exec.c	isinf.c	qsort.c	strtol.c
fseeko.c	kill.c	rand.c	strtoul.c
getaddrinfo.c	memcmp.c	random.c	thread.c
gethostname.c	noblock.c	rint.c	unsetenv.c
getopt.c	open.c	snprintf.c	
getopt_long.c	path.c	sprompt.c	

Build Environment

- adjust Makefiles and build
- errno vs. GetLastError()

Kernel

- IPC - memory, semaphores
- sleep interrupted
- getrusage()
- memory allocation from shared libraries

Miscellaneous

- user name lookups, no /etc/passwd
- locale vs code page
- no accessible timezone database
- relocatable installs
- Win32 service
- installer MSI

Conclusion

