



KDE4 on Solaris

KDE Everywhere

Adriaan de Groot

November 26, 2008



KDE4 on Solaris

Tools and Techniques for Portability

Adriaan de Groot, KDE e.V.



Who is this guy?



- Researcher in Software Quality
 - Radboud University Nijmegen (Verification)
 - Athens University of Economics and Business (SQO-OSS)
- Board Member KDE e.V. and NLUUG
- Two kids and a lot of Lego bricks



Who is this guy? (2)



- First “Free” Software contribution in 1989
- KDE since 2000
 - PIM, Handheld Sync
 - FreeBSD ports
 - Code quality checking
 - OpenSolaris ports
 - Legal portfolio



What is Solaris?



Solaris

- Proprietary UNIX; SysV derived
- Strong adherence to standards (POSIX.1, SUS, XPG)
- Annoyingly different

OpenSolaris

- Free Software UNIX; SysV derived
- Strong adherence to standards (POSIX.1, SUS, XPG)
- *Still* annoyingly different



What is KDE?



- Free Software desktop
- Cross-platform
 - Historically “all the UNIXes”
 - Now Linux, FreeBSD, Windows, MacOSX



What is KDE? (2)

- D-BUS for IPC
- HAL for hardware
- GStreamer for audio (*)
- XDG standards (*)



Belenix!



- Project started November 2007
- Produce packages for KDE4 on (Open)Solaris
 - “Pristine” upstream
 - Using Sun Studio 12
 - On S10U5 + OSOL, amd64 + SPARC, 32 + 64 bit
 - Complete KDE4 desktop experience
- Goal: KDE4 IPS packages in OSOL 2009.4



The Team

- Stefan Teleman, Lukas Oboril, Adriaan de Groot
- Working directly with upstream
- Somewhat supported by Sun

Contacts

- Edward O'C, Ben Taylor
- Joep Vesseur, Gerard van den Berg
- Roman Shaposhnik



Why Bother?

- Well, it *is* Free Software
- CDE rocks!
- Customer demand
- Technical challenge
- .. and it's good for KDE



- Exposes Linuxisms
- Exposes gccisms
- Exposes C++ issues
- Reciprocal: Studio bugs
- Reciprocal: Solaris bugs



Then ...

All the world's a VAX
(32-bit, little-endian, CISC, DEC UNIX)

Now ...

All the world's a PC
(32-bit, little-endian, CISC, Linux)

And in between ...

32-to-64 bit conversion; big endian PPC; RISC machines;
proprietary UNIXes



/proc

```
char pathbuffer[MAXPATHLEN + 1];  
pathbuffer[MAXPATHLEN] = 0;  
int length = readlink ("/proc/self/exe",  
pathbuffer, MAXPATHLEN);
```



- `/usr/include`
- `ioctl`s `TIOCOUTQ` vs `FIONREAD`
- `“GLIBC” == “linux`
- Integer types `uint32_t` vs. `u_int32_t`
- Traditional portability bugbears



With a good framework / abstraction,
this *should not* matter.



- Lax interpretation of standards
- Own STL
- Extensions to C++



```
class A { A() { } } ;  
void foo() {  
  A a = A::A();  
}
```



```
class A { A() { } } ;  
void foo() {  
  A a = A();  
  // A a;  
}
```



```
class A { void foo(const int i); } ;  
void A::foo(int i) { }
```



```
class A { void foo(const int i); } ;  
void A::foo(const int i) { }
```



```
int a;  
float b = abs(a);
```



- Variations in the STL
 - Sun Cstd based on 1995 SGI STL
 - g++ STL based on 1996 STL
- OpenSolaris STL to be updated 2009



- UBE crashes, optimizer crashes ...
- `int foo() { if (0) return 1; }`
- ... and a good working relationship.



- Missing 64-bit libraries
- Broken python header file



- Keeping up with Linux
- Introducing equivalent functionality
- Integrating with the host system
 - Zones, Containers
 - ZFS support



System Information (Solaris)

```
/* traverse the kstat chain to find the appropriate statistics */  
if( (ksp = kstat`lookup( kctl, "unix", 0, "system`pages" )) ==  
NULL )  
return( 0 );  
if( kstat`read( kctl, ksp, NULL ) == -1 )  
return( 0 );
```

System Information (Linux)

```
if ( ( fd = open( "/proc/meminfo", O`RDONLY ) ) < 0 )  
return -1;  
n = read( fd, MemInfoBuf, MEMINFOBUFSIZE - 1 );
```



- Blastwave
- Belenix



- Integration of stdcxx and Qt
- Continued push of dependencies
- Proposal: a Linux-style interaction model
 - KDE Project does the Solaris basics
 - Belenix does the system integration
 - Upstreaming efforts