

SCO Forum 2006

MOBILITY EVERYWHERE >



Presentation Title: Developing for SCO UNIX Platforms

Presenter's Name: John Wolfe & Jonathan Schilling

Session ID: Tues: 3:45 PM - 126

Wed: 9:45 AM - 137

1



Platinum Sponsor



Get Your Passport Stamped



- Be sure to get your Passport stamped.
 - Get your passport stamped
 - By breakout session instructors
 - By exhibitors in the exhibit hall
 - Turn in your Passport
 - After the last breakout session on Wednesday
 - Drawing for great prizes for Wrap-up Session
- Remember to complete the breakout session evaluation form, too

WIN BIG

SCO Forum 2006
PASSPORT

Turn in this card at the Registration and Information desk. Prize drawings will be held during the Closing Session of SCO Forum, at 4pm on Tuesday, August 9th. You must be present to win.

HOW
> Att
> Visi
> Hav
Atten
a drap
iPods

Name: _____
Company: _____
eMail: _____
Phone: _____

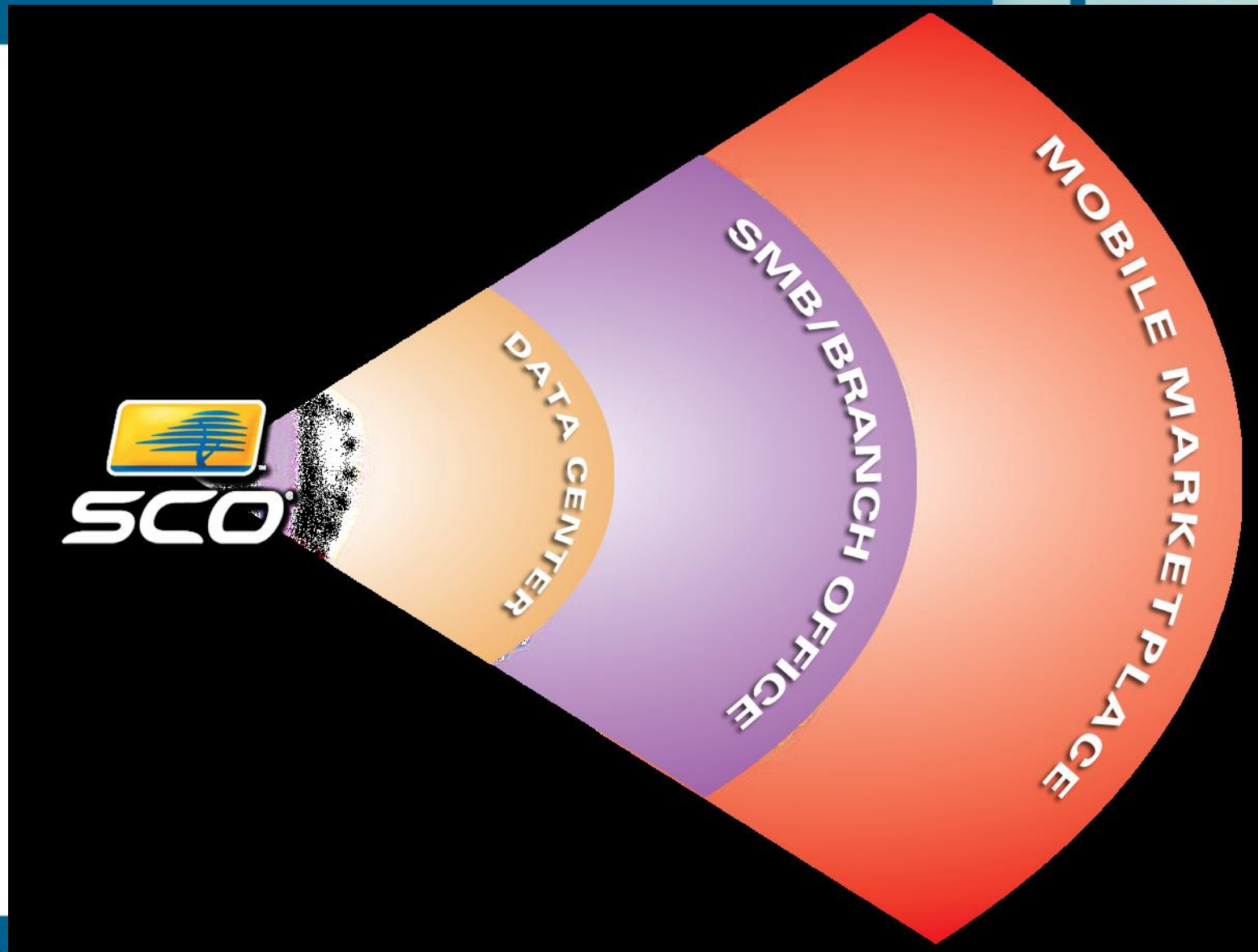
Breakout Sessions

Monday: ○ ○ ○
Tuesday: ○ ○ ○

Tradeshow

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

SCO Automates Transactions



SCO UNIX Products



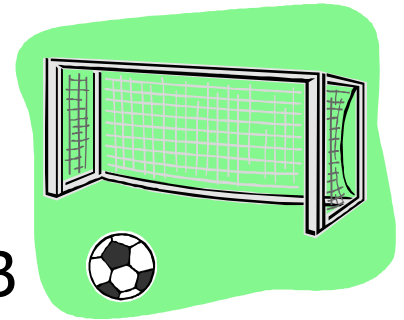


- In this session we will cover:
 - The structure of SCO OpenServer 6 and its Devsys
 - How to single certify applications for it
 - How to modernize OpenServer 5 applications for it
 - Status of Java and SCO Web Services
 - Features of the OpenServer 6 (and UW7) Devsys
 - With emphasis on:
 - **debug** tips and tricks for Dbx or GDB users
 - **memtool** to find application memory leaks and corruption
 - **fur** to improve application performance
 - Resources for developers

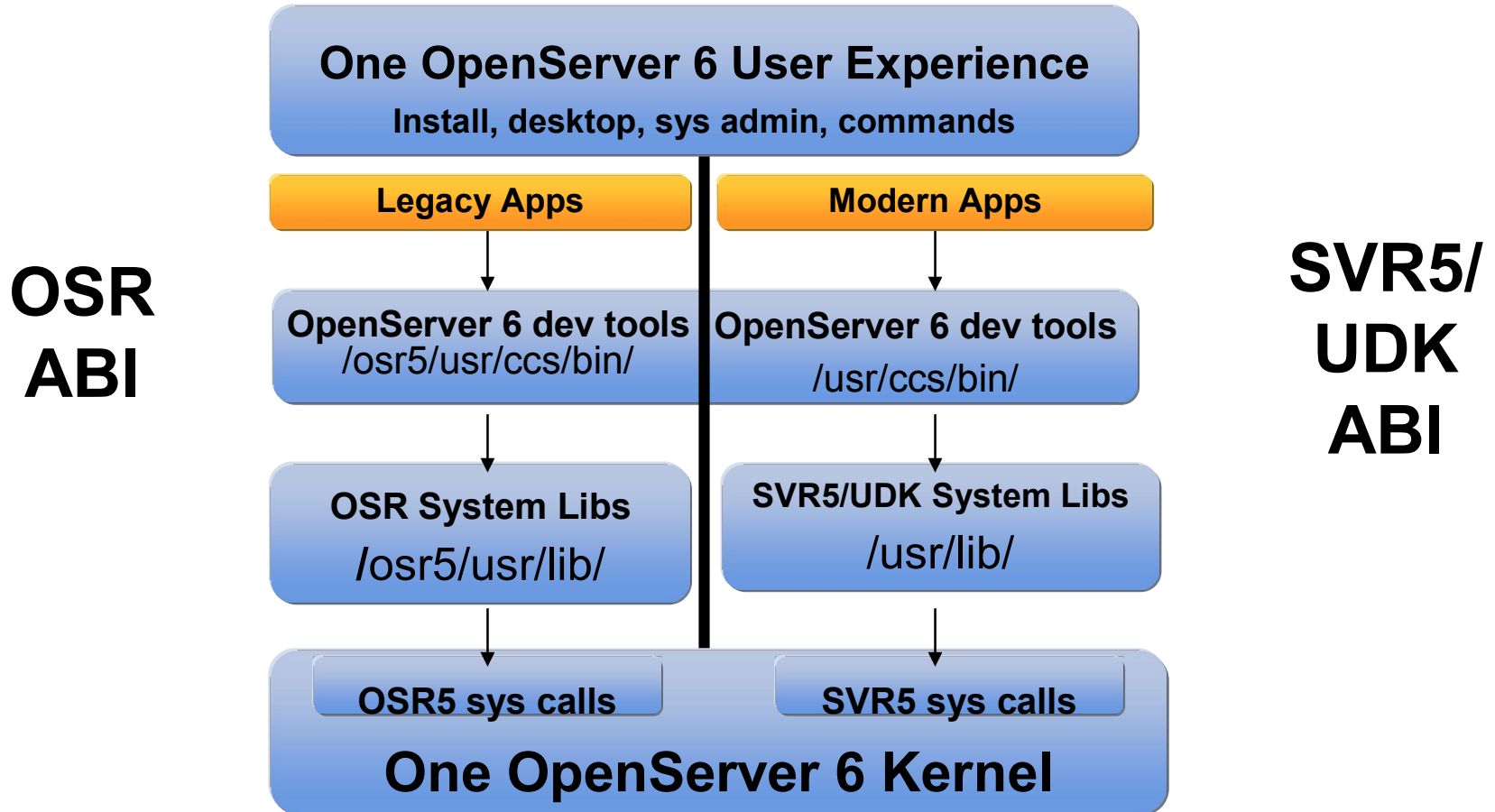
Goals of OpenServer 6



- A modernization release of SCO OpenServer 5
 - adds kernel threads
 - adds user-space POSIX threads library
 - adds asynchronous I/O
 - adds LFS large file systems for files > 2 GB
 - better performance, reliability, scalability
 - adds Java 1.4.x and Java 2 SE 5.0 (in progress)
 - enables threads-dependent apps
 - Apache 2, MySQL, KDE desktop, etc.
 - **user interfaces - still the same as OpenServer 5!**
 - **install, desktop, sys admin, commands, etc.**



How OpenServer 6 is Structured





- OpenServer 6 Dev Sys for SVR5 (UDK) ABI
 - OpenServer 6 dev sys using **-K udk** option
 - or - **/usr/bin/cc** which defaults to **-K udk**
 - use for single certification on UnixWare 7 and OpenServer 6
 - use for modernizing existing OSR5 apps
 - use for device driver development (IHVs)
 - used to relink the OpenServer SVR5 kernel
 - provides access to "NEW" features
 - threads and LFS (> 2 Gbyte files)

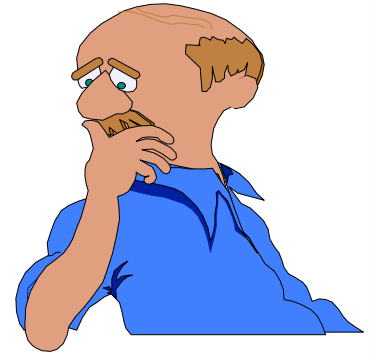


- OpenServer 6 Dev Sys for OSR ABI
 - OpenServer 6 dev sys using **-K osr** option
 - or **/osr5/usr/bin/cc** which defaults to **-K osr**
 - set PATH environment variable with **/osr5/usr/bin** before **/bin, /usr/bin** or **/usr/ccs/bin**
 - use for binary compatibility to legacy OSR5 apps
 - provides more modern C and C++ compilers
 - Standards Conformance (almost) C and C++
 - same level of code generation and optimization as in the SVR5/UDK compilers
 - 64-bit "long long"

So exactly what is an ABI?



- What an app looks like at the binary level
 - content and layout of information that it presents to system loaders and linkers (object file format)
- How different modules of an app communicate
 - function call conventions
 - size and layout of basic data types
 - size and layout of compound data types - structures, unions, bit-fields
- How an app communicates with the system
 - pathnames, sys call numbers, errno's, ioctls
 - size and layout of basic and aggregate system data types





- No safe way to link OSR5 ABI and SVR5/UDK ABI relocatable or shared objects (.o/.a/.so)
 - no way to intercept different system data types
 - no way to intercept different bit-field layouts and function calling conventions
 - no way to intercept system calls from objects
- Linker will reject mixture of objects, by default
- Force link mode provided - "I know what I'm doing"
 - but you probably don't
 - not recommended



Want new features but need compatibility with old OSR5 library?



- If your own, recompile
- If from another ISV, get vendor to provide new, SVR5 ABI-built libraries
- If neither is possible ...
 - make app into 2 processes
 - one process calls old lib
 - compile **-Kosr**
 - one process uses new features
 - compile **-Kudk**
 - use socket, pipe, IPC, etc. to communicate between processes



Single Certification for UW7 and OSR6



- Major advantages for ISVs
 - one build environment
 - one binary distribution
 - one platform for full test cycle
 - two platforms it can run on
 - two markets it can sell to
- SVR5 (UDK) is the vehicle
- What are the Do's and Don'ts?

Single Certification Platforms



- Supported now
 - develop on UnixWare 7.1.x, run on OpenServer 6
- Supported in VERY-NEAR future
 - develop on OpenServer 6, run on UnixWare 7.1.4 MP4
 - Xserver (X.org) and graphics (GWXlibs) from OpenServer 6 are being added UW7.1.4 MP4

Single Certification Testing Guidance



- Really only need to test once?
- Do full certification testing on one platform
- Do sanity checks/touch testing on the other platform
 - treat similarly to re-testing product on a new minor point release of an OS
 - we *expect* it to work
 - if it doesn't, it's a bug and we'll fix it!
- First time you do this, do some extra testing on the second platform



- General rules
 - link apps dynamically, not statically
 - use threads, LFS, networking, graphics, audio, almost all other APIs!
 - avoid certain auditing/security/admin/job control APIs
 - use C, C++, Java, or assembly
 - use APIs to get system information
 - don't read and parse **/etc** and **/var** files directly
 - use pkgadd format
 - use shells/commands that behave the same
 - or use "non-native" shell/commands on one platform



- APIs
 - all “useful” APIs from normal applications, work
 - Thousands!
 - Authentication APIs:
 - use the SVR5 (UW7) libiaf APIs
 - - Available and working in MP2



- Commands
 - shells are somewhat different across the ABIs
 - e.g., OSR5 sh doesn't support symbolic signal names
 - use **/u95/bin/sh** instead of **/bin/sh**
 - now will get consistent behavior across UW7, OSR6
 - use uname to determine which platform on
 - use **/udk/bin** and **/u95/bin** on OSR6 to find commands with same behavior as UW7
 - prefix your PATH setting with these



- System directory structure and file differences
 - apps should not read and parse **/etc** and **/vars** configuration files directly
 - apps should use APIs instead
 - example: reading **/etc/inet/hosts** works on UW7 but not on OSR6 where that file doesn't exist
 - call `gethostent()` etc., works on both UW7 and OSR6
 - **This is good portability practice anyway!**
 - device names should be configurable
 - don't hard-code inside application
 - many UW7 device names are present in OSR6 anyway
 - e.g. `/dev/tty00s`, `tty01s`, etc. for serial lines



- Packaging
 - packages should install into **/opt** or **/var**
 - with two-ABI world, installing into /usr is ill-advised
 - use pkgadd technology – if using SCO packaging tools
 - available on both UW7 and OSR6
 - custom is only on OSR6
 - some ISVs have their own install technology
 - GUI-driven install wizards
 - simple tarballs

Single Certification for Drivers



- UnixWare 7 device drivers work on OpenServer 6
 - It's the same SVR5 kernel technology
- IHV Single Certification between UnixWare 7 and OpenServer 6 (either direction) is easy and problem-free
- Use pkgadd format !

Single Cert. for Drivers: Examples



- Adaptec HostRAID SAS driver (adp94xx)
 - available now as alpha - FCS 10/1/2006
 - Adaptec Storage Manager (Java app)
- LSI MegaRAID SAS (megasas)
 - available now as alpha - FCS 10/1/2006
 - command line management tools by FCS
- LSI MPT-Fusion (mpt)
 - SAS support added in version 1.04.10
 - available from LSI



- Existing OSR5 DevSys C++ compiler is old!
 - AT&T Cfront-based, c. 1992, buggy
 - predates 1998 ISO language/library standards
 - large-scale changes in language since then
- If your sources were developed with it ...
 - expect they will *not* compile cleanly now
 - source fix-ups are usually straightforward
 - you're doing your code a favor!
 - for bad cases try the CC -Xo option
 - old library classes will all still be there

Guidance on modernizing existing apps: Threads



- Must modify to use threads
 - pthreads API more standard than SVR4/UI threads
 - use **-Kudk** to recompile application
 - use **-Kpthread** when compiling threaded code
 - fixes some things like global errno automatically
- Existing OSR5 source may not be thread-safe!
 - may use non-reentrant functions such as **strtok()**
 - use **strtok_r()** replacements when available
 - may store application data globally
 - may return pointers to static data
 - must study your code



- Go forward with Large File Summit (LFS) APIs
 - use **-Kudk** to recompile application
 - create files up to one terabyte in size
 - can use size-specific interfaces
 - **fopen64, lseek64**, etc.
 - or, can use regular **fopen, lseek**, etc.
 - **cc -D_FILE_OFFSET_BITS=64**
 - must use vxfs filesystem and create filesystem with **largefiles** flag
 - **mkfs** or **fsadm_vxfs** to turn on/off
 - **ulimit** must be set to **unlimited**

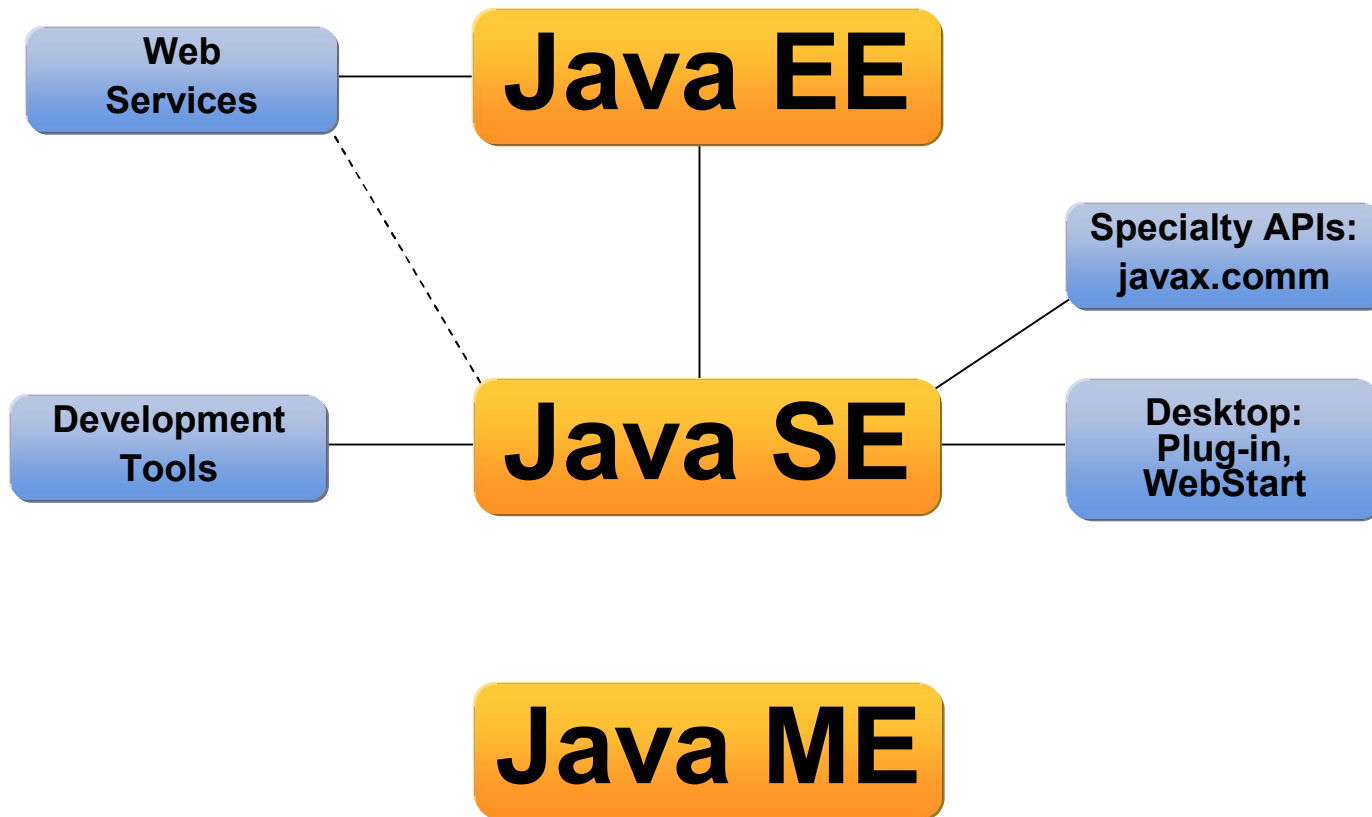


- When is OSR ABI needed?
 - when linking with existing OSR5 .o/.a/.so objects
- Use OSR ABI compilers
 - same as UDK but with **-Kosr** for OSR ABI
 - modern, reliable, standard, optimizing
 - 64-bit “long long” integer available
 - LFS, threads, asynch I/O, EFT *not* available
 - can accept OSR5 COFF objects as input to linker
 - but cannot *generate* COFF
 - can link with existing OSR5 C .o/.a/.so objects
 - but *cannot* link with existing OSR5 C++ objects
 - use **CC -Xo** to compile very old OSR5 C++ sources



- Be careful with expanded fundamental system types (EFT)
 - Size change between OSR5 and OSR6 in UDK mode:
 - mode_t, dev_t, uid_t, gid_t, nlink_t, pid_t, ino_t, sigset_t
 - typically size goes from 16 bits to 32 bits
 - system or app struct's containing them also change size
 - e.g., struct stat contains both dev_t and ino_t
 - dev_t also changes how major, minor numbers packed
 - all consequences of SVR5 infusion into OpenServer 6 kernel
 - Change should be transparent unless your code has assumptions about size

The Pieces of Java





- **SCO OpenServer 6.0.0**
- **UnixWare 7.1.4**
- **UnixWare 7.1.3 Update Pack 4**
 - essentially equivalent to UW 7.1.4
- **SCO OpenServer 5.0.7 MP3/UP3/SuppCD3**
 - requires OSRcompat v. 8.0.2 or higher
 - uses new user-space OSR5 UDK threads library

Native Threads on OpenServer 5 !?



- SCO OpenServer 5 UDK threads library
 - UW7 libthread ported to OSR5 under UDK
 - “available LWPs always equals one” model
 - required for Java 1.4.2
 - “green threads” are gone
 - also use for UDK C/C++ apps
- SCO OpenServer 5 ABI threads library?
 - no, this is UDK only
 - not needed for Java



Here now !!!!



- J2SE 5.0 – update 6
 - available at the end of August, 2006
 - acquiring permission to distribute various CA Root Certificates
- J2SE 5.0 – update 8
 - available in September, 2006
 - scheduled security fixes



The moral is ...



- If you're using Java on an older OSR5
 - **Upgrade to OSR6!**
- If you're using Java on an older UW7
 - **Upgrade to UW7.1.4!**
- Java 1.4.2 will *not* be backported to earlier releases
- Java 1.5 will *not* be made available on back releases

Java EE — “Enterprise Edition”



- a/k/a “Java app server”
- The **Java middleware standard** for servers
- For many end user app builders and ISVs, this is what you program to
- Many vendors
- JDBC (Database)
- Enterprise Java Beans
- JavaServer Pages
- Servlets
- Web Services, XML
- JNDI (Naming & Directory)
- CORBA
- Java Message Service
- Transactions
- and much more





- Proprietary
 - BEA WebLogic
 - IBM WebSphere
 - Pramati
 - various others
- Open Source
 - JBoss
 - Sun Glassfish
 - See Sun's Java EE presentation here at Forum!
 - Apache Geronimo
 - ObjectWeb JOnAS



- “Enterprise” features not just for “Enterprises”
 - useful for SMB and replicated sites too
- BEA WebLogic Server 8.1
 - BEA’s most recent release
 - certified for native UnixWare 7.1.4 and Java 1.4.2
- Open source implementations
 - just download and run, no porting usually necessary
 - UW7 users report good success with JBoss
 - Glassfish should be similar
- Don’t always need the whole enchilada!
 - for just servlets & JSP, use Tomcat package
 - JDBC usually comes from database vendor
 - for SOAP/XML web services, see later on ...

Java Serial and Parallel I/O Support



- javax.comm API is an extension to Java
 - RS-232 based serial I/O
 - IEEE 1284 based parallel I/O
- Available on OSR 5.0, 7, OSR6 mp1, and UW 7.1.4
 - based on RXTX open source implementation
 - a good number of users on OSR5
 - see </usr/java/javax.comm-ReleaseNotes.html> for guidelines on how to use



- Parallel I/O guidelines
 - supports basic printing
 - status APIs not supported
 - isPaperOut, isPrinterBusy, isPrinterError, etc.
- USB support?
 - full-blown USB support is a different API
 - javax.usb (JSR 80)
 - but behave-like-serial *will* work with javax.comm
 - may depend on the actual driver
 - success stories on OSR5



- Web Services – What are they?
 - provide layer of software abstraction
 - make apps available via Internet to other apps
 - OS, hardware, language, location agnostic
 - industry-standard specifications
 - SOAP, XML, WSDL, UDDI
 - Don't need to know, tools auto generate
 - bridge between J2EE and .NET
 - ideal for writing EdgeClick "agents"



- Three operating systems
 - OSR 6.0.0 mp1
 - OSR 5.0.7 supp3
 - UW 7.1.4
- Two web service servers
 - Apache
 - Tomcat
- Five languages
 - Java, Perl, PHP included in each's default set of libraries
 - C, C++ at `/usr/lib/gsoap/`
 - Demos of client use in `/usr/scox/language_demos/`

SCOx Web Services Language Support



SOAP LIBRARY

XML PARSERS

SECURITY

C / C++	gSOAP 2.2.3	expat libxml	OpenSSL
Java	Apache Axis 1.2 w/ fix	Apaches Xerces-J 1.4.4	JSSE within J2SE 1.4.2
Perl	SOAP::Lite	XML::Parser	IO::Socket::SSL Crypt::SSLeay
PHP	PEAR::SOAP	PHP XML (expat) PHP DOM XML (libxml)	OpenSSL curl

Why Java fits in with Web Services

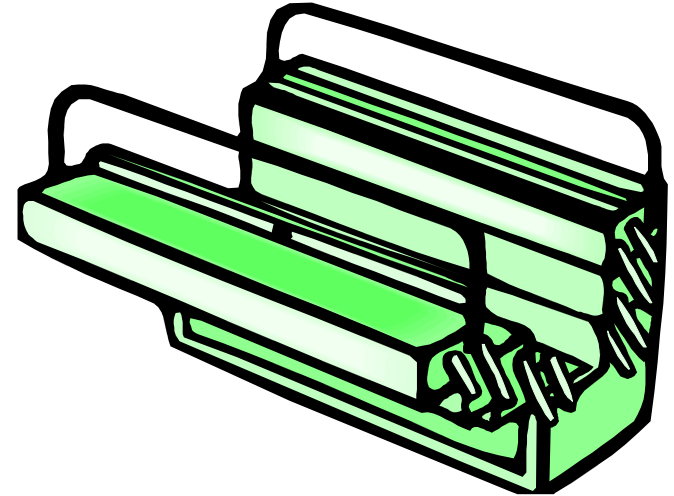


- Java is an excellent language for Web Services
 - part of J2EE 1.4 specification
 - can also use stand-alone apart from J2EE
- Language is flexible, dynamic, self-aware
 - proxy classes, complex types, XML data binding ...
- Java web services bindings are **standards**
 - JAXP, JAXB, JAX-RPC, etc.
 - not true of other languages (C, C++, PHP ...)
- Java encourages powerful middleware layers

Features of the OpenServer 6 Dev Sys



- C Compilation System
- C++ Compilation System
- C/C++ Debugger
- memtool
- fur
- Except where noted, features apply to Dev Sys used for both SVR5/UDK and OSR ABIs and to UDK on UW7
- A major upgrade compared to existing (and outdated) OSR5 Development System product!!



The OpenServer 6 -K mode switch



- Compilers
 - **/usr/ccs/bin/cc** defaults to **-Kudk**
 - **/osr5/usr/ccs/bin/cc** defaults to **-Kosr**
 - “cross-ABI” compiles are allowed
 - **/usr/ccs/bin/cc -Kosr ...**
 - **/osr5/usr/ccs/bin/cc -Kudk ...**
 - ditto CC for C++ compiles – as & ld also
 - Use cc or CC to do linking – links against correct ABI startup routines.
- Other Dev Sys commands
 - have **-K osr | udk** option if necessary (e.g. lint)
 - don't have option if irrelevant (e.g. lex and yacc)

OpenServer 6 C Compiler



- Robust compiler, excellent IA-32 code generation
- Standards-conforming libraries and headers
- Profiled versions of libraries
 - `prof`, `lprof` in both ABIs
 - `fprof` SRV5/UDK ABI only
- Standard set of command line tools, fully internationalized
- Conformance checking (`-Xc`) is against C 90 standard
- Support for Java native methods [SVR5/UDK ABI only]
- Almost all of C 99 - ISO/IEC 9899:1999
 - `inline`, `restrict`, variable argument macro functions, & 60 other features
 - Only things missing:
 - variable-length arrays
 - complex and imaginary numbers
 - minor variances in `snprintf(3S)`
 - [some new C99 library functions and headers may be SVR5/UDK ABI only]
 - Option `-Xb` will disable `inline` and `restrict` keywords

OpenServer 6 C++ Compiler



- Accurate, robust implementation
- Almost all of the C++ standard - ISO/IEC 14882:1998
 - except rarely-used: export keyword, placement delete, function-try-blocks, two-phase template name binding, multi-byte characters in source code, partial specialization of a class member template
- Complete C++ Std Library
 - STL, iostreams, string, locale, numerics, etc.
 - fast and thread-safe
- Excellent IA-32 code generation
- Exception Handling - high performance
- Device driver support
- Thread safety [SVR5/UDK ABI only]
- Support for Java native methods [SVR5/UDK ABI only]

OpenServer 6 Debugger - debug



- Graphical user interface
 - User configurable screen layouts
- Command line interface
 - powerful, shell-like command language
 - command history, command aliases
- Strong C and C++ symbolic debugging
 - Step through inline functions, header code, exceptions
- Controls multi-process, multi-threaded apps
 - follow forks in both parent and children processes
- Understands ELF/COFF, DWARF I/II executables



debug - How to get started



- **debug** – man page
- **help** command in the debugger
 - **help** – lists available commands and topics
 - **help <cmd-topic>** - format and details about command or topic
- Use the on-line/locally installed debugger doc. “Debugging and analyzing C & C++ Programs”
 - Command line and GUI
 - Tutorials, explanations, and tips

debug - For dbx or gdb users



- **dbx** users

- Section 3 of the Porting Guide "A Guide to debug for dbx Users"
 - URL provided at end of presentation

- **gdb** users

- command comparisons from May/June 2000 SCO World article
 - attached to the back of the handout

debug - Personal Configuration



- **alias** command
 - define alternate / abbreviated commands
 - use to establish dbx-like or gdb-like commands
 - build complex, repetitive, conditional command sequences
- **\$HOME/.debugrc**
 - startup **debug** command script
 - establish *my_former_debugger*-like configuration
 - **debug ... -d <alt_startup> ...**
 - uses specific alternate startup script instead of default

debug - logon / logoff / script



- **logon <log_file>**
 - logs debug commands entered and output to a file
 - generated output appears as comments
 - capture complete history
 - capture repetitive command sequence
- **logoff**
 - Terminate logging
- **script <file>**
 - reads debug commands from <file>

memtool - Catching Dynamic Memory Errors



[SVR5/UDK ABI only]

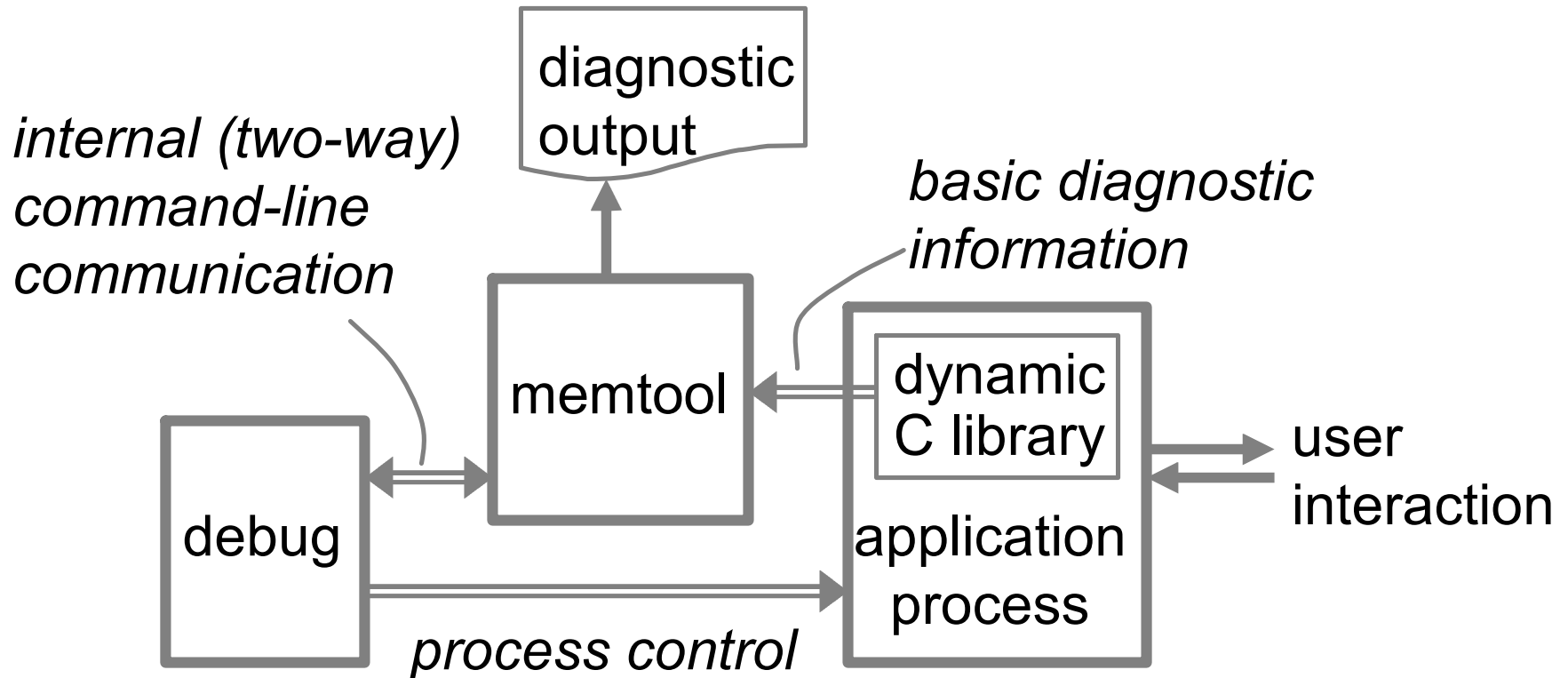
- Diagnose dynamic memory allocation errors
 - writing beyond a block of memory
 - using deallocated blocks
 - memory “leaks”
 - bad arguments passed to C **malloc** or C++ **new**
- Does not catch general pointer misuses or writing outside local or global arrays
- Runs the application under the hidden control of the debugger and the dynamic C library **malloc** runtime checking





- Diagnostics include one to three stack traces
 - when detected
 - when (de)allocated
 - previous use (for **realloc** or **free**)
- Erroneously modified block diagnostics include an annotated memory dump snapshot for the block
- Each diagnostic comes with an explanation – short, medium, or long (user selectable)
- Application need not be rebuilt or relinked
 - debugging (**-g** flag) provides much better info

memtool – Under the covers



memtool – Example (overly simple)



```
$ cat bad.c
#include <stdlib.h>

int main(void) {
    char *p = malloc(13);
    p[13] = 'x';
    return 0;
}

$ cc -g -o bad bad.c
$ ./bad
```

Note: no errors noticed at runtime

memtool – Example (continued)



Much more than the following will be displayed by memtool for this example. This is intended to give you a feel for what memtool's output looks like.

```
$ memtool ./bad
```

```
A block's spare bytes have been modified. This usually occurs due to writing beyond the block's regular bytes, either because an insufficient number of bytes were requested when the block was allocated or simply due to a programming logic error.
```

```
*Stack trace when block at 0x8049660 was allocated:
```

```
[0] malloc(sz=13)
```

```
[1] main() [bad.c@3] in ./bad
```

```
[2] _start(presumed:0x1,0x8047a74,0x8047a7c) [0x804850d] in ./bad
```

fur - Function Relocation/Reorganization



- Instruments and reorders code in relocatable objects - at code block level
 - changes function order - locality of reference
 - reorder code blocks within functions - improves branch prediction
 - move "low-usage" code to pseudo functions
- Profile guided optimization without recompilation
- Actual uses
 - Has been used to optimize SCO kernels and libraries
 - OEMs used fur to gain 15% improvement in Oracle DB server - TPCC benchmarks
 - Used for J2SE and soon for Firefox[®] browser

fur – Process Block profiling [1]



1. Compile and partially link as relocatable object
 - for C++, must prelink before linking: **-Tprelink_objects**
 - **ld -r -o xxxx.rel** to link – no runtime libraries
 - save a copy of the **xxxx.rel**
2. Use fur to instrument every basic block of the **xxxx.rel**
 - **fur -b [all | flow] -K <keep_file> **
-c "mkblocklog -p <log_dir>/block.xxxx" xxxx.rel
3. Complete the link producing an instrumented a.out
 - **cc/CC ... xxxx.rel log.xxxx.o**
4. Run representative workloads
 - multiple runs creates consecutive block.xxxx.??



5. Analyze all the log files, creating optimal reordering file
 - copy saved xxxx.rel to xxxx.rel
 - **fur -r -o <order_file> -k <keep_file> \
<log_dir>/block.xxxx.01,<log_dir>/block.xxxx.02,... -m
xxxx.rel**
6. Apply the reordering file against the **xxxx.rel** and complete the FINAL link
 - **fur -o <order_file> -k <keep_file> xxxx.rel**
 - **cc/CC xxxx.rel**
- Once <order_file> is obtained and preserved, only steps #1 and #6 need be done on subsequent rebuilds
 - If function shape changes, fur will NOT reorder that function
 - Heavy source changes – repeat all steps to produce “current” <order_file>

OpenServer 6 Support Resources

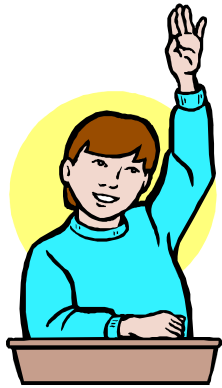


- Porting Guide:
 - <http://www.sco.com/support/docs/openserver/600/porting/osr6portingTOC.html>
- Upgrade Guide:
 - <http://www.sco.com/support/docs/openserver/600/upgrade/index.html>
- Online Documentation and Late News
 - <http://www.sco.com/support/docs/openserver/>

OpenServer 6 Support Resources



- Support Download Page for OpenServer 6:
 - <http://www.sco.com/support/update/download/product.php?pfid=12&prid=20>
- SCO "Legend" Mailing List: **Public**
 - Legend-subscribe@list.sco.com
- Porting/Migration Alias:
 - osr5to6@sco.com
- Knowledge base:
 - <http://wdb1.sco.com/kb/search>



Questions



Get Your Passport Stamped



- Be sure to get your Passport stamped.
 - Get your passport stamped
 - By breakout session instructors
 - By exhibitors in the exhibit hall
 - Turn in your Passport
 - After the last breakout session on Wednesday
 - Drawing for great prizes for Wrap-up Session
- Remember to complete the breakout session evaluation form, too

WIN BIG

SCO Forum 2006
PASSPORT

Turn in this card at the Registration and Information desk. Prize drawings will be held during the Closing Session of SCO Forum, at 4pm on Tuesday, August 9th. You must be present to win.

HOW
> Att
> Visi
> Hav
Atten
a drap
iPods

Name: _____
Company: _____
eMail: _____
Phone: _____

Breakout Sessions

Monday: ○ ○ ○
Tuesday: ○ ○ ○

Tradeshow

○ ○ ○ ○ ○ ○ ○ ○ ○ ○