BSDPort hotspot shared changes, cpu

Here are the proposed jdk8 hotspot/src/cpu changes:

```
diff -r e6e7d76b2bd3 src/cpu/sparc/vm/globals_sparc.hpp
--- a/src/cpu/sparc/vm/globals_sparc.hpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/sparc/vm/globals_sparc.hpp        Mon Jul 25 17:04:06 2011 -0700
@@ -70,7 +70,11 @@
 define_pd_global(bool, RewriteBytecodes,     true);
 define_pd_global(bool, RewriteFrequentPairs, true);

+#ifdef _ALLBSD_SOURCE
+define_pd_global(bool, UseMembar,            true);
+#else
 define_pd_global(bool, UseMembar,            false);
+#endif

 // GC Ergo Flags
 define_pd_global(intx, CMSYoungGenPerWorker, 16*M);  // default max size of CMS young gen, per GC worker thread
diff -r e6e7d76b2bd3 src/cpu/x86/vm/bytes_x86.hpp
--- a/src/cpu/x86/vm/bytes_x86.hpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/bytes_x86.hpp        Mon Jul 25 17:04:06 2011 -0700
@@ -81,6 +81,9 @@
 #ifdef TARGET_OS_ARCH_windows_x86
 # include "bytes_windows_x86.inline.hpp"
 #endif
+#ifdef TARGET_OS_ARCH_bsd_x86
+# include "bytes_bsd_x86.inline.hpp"
+#endif


 #endif // CPU_X86_VM_BYTES_X86_HPP
diff -r e6e7d76b2bd3 src/cpu/x86/vm/c1_LIRAssembler_x86.cpp
--- a/src/cpu/x86/vm/c1_LIRAssembler_x86.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/c1_LIRAssembler_x86.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -480,8 +480,8 @@
   // Fetch the exception from TLS and clear out exception related thread state
   __ get_thread(rsi);
   __ movptr(rax, Address(rsi, JavaThread::exception_oop_offset()));
-  __ movptr(Address(rsi, JavaThread::exception_oop_offset()), (int32_t)NULL_WORD);
-  __ movptr(Address(rsi, JavaThread::exception_pc_offset()), (int32_t)NULL_WORD);
+  __ movptr(Address(rsi, JavaThread::exception_oop_offset()), (intptr_t)NULL_WORD);
+  __ movptr(Address(rsi, JavaThread::exception_pc_offset()), (intptr_t)NULL_WORD);

   __ bind(_unwind_handler_entry);
   __ verify_not_null_oop(rax);
diff -r e6e7d76b2bd3 src/cpu/x86/vm/copy_x86.hpp
--- a/src/cpu/x86/vm/copy_x86.hpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/copy_x86.hpp        Mon Jul 25 17:04:06 2011 -0700
@@ -37,6 +37,9 @@
 #ifdef TARGET_OS_ARCH_windows_x86
 # include "copy_windows_x86.inline.hpp"
 #endif
+#ifdef TARGET_OS_ARCH_bsd_x86
+# include "copy_bsd_x86.inline.hpp"
+#endif


 static void pd_fill_to_words(HeapWord* tohw, size_t count, juint value) {
diff -r e6e7d76b2bd3 src/cpu/x86/vm/globals_x86.hpp
--- a/src/cpu/x86/vm/globals_x86.hpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/globals_x86.hpp        Mon Jul 25 17:04:06 2011 -0700
@@ -70,7 +70,11 @@
 define_pd_global(bool, RewriteBytecodes,     true);
 define_pd_global(bool, RewriteFrequentPairs, true);

+#ifdef _ALLBSD_SOURCE
+define_pd_global(bool, UseMembar,            true);
```

```
+#else
 define_pd_global(bool, UseMembar,            false);
+#endif

 // GC Ergo Flags
 define_pd_global(intx, CMSYoungGenPerWorker, 64*M);  // default max size of CMS young gen, per GC worker thread
diff -r e6e7d76b2bd3 src/cpu/x86/vm/interp_masm_x86_32.cpp
--- a/src/cpu/x86/vm/interp_masm_x86_32.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/interp_masm_x86_32.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -45,6 +45,9 @@
 #ifdef TARGET_OS_FAMILY_windows
 # include "thread_windows.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif


 // Implementation of InterpreterMacroAssembler
@@ -1144,7 +1147,7 @@
   int recvr_offset = in_bytes(VirtualCallData::receiver_offset(start_row));
   set_mdp_data_at(mdp, recvr_offset, receiver);
   int count_offset = in_bytes(VirtualCallData::receiver_count_offset(start_row));
-  movptr(reg2, (int32_t)DataLayout::counter_increment);
+  movptr(reg2, (intptr_t)DataLayout::counter_increment);
   set_mdp_data_at(mdp, count_offset, reg2);
   if (start_row > 0) {
     jmp(done);
@@ -1287,7 +1290,7 @@
     test_method_data_pointer(mdp, profile_continue);

     // Build the base (index * per_case_size_in_bytes()) + case_array_offset_in_bytes()
-    movptr(reg2, (int32_t)in_bytes(MultiBranchData::per_case_size()));
+    movptr(reg2, (intptr_t)in_bytes(MultiBranchData::per_case_size()));
     // index is positive and so should have correct value if this code were
     // used on 64bits
     imulptr(index, reg2);
diff -r e6e7d76b2bd3 src/cpu/x86/vm/interp_masm_x86_64.cpp
--- a/src/cpu/x86/vm/interp_masm_x86_64.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/interp_masm_x86_64.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -45,6 +45,9 @@
 #ifdef TARGET_OS_FAMILY_windows
 # include "thread_windows.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif


 // Implementation of InterpreterMacroAssembler
diff -r e6e7d76b2bd3 src/cpu/x86/vm/jni_x86.h
--- a/src/cpu/x86/vm/jni_x86.h        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/jni_x86.h        Mon Jul 25 17:04:06 2011 -0700
@@ -26,7 +26,7 @@
 #ifndef _JAVASOFT_JNI_MD_H_
 #define _JAVASOFT_JNI_MD_H_

-#if defined(SOLARIS) || defined(LINUX)
+#if defined(SOLARIS) || defined(LINUX) || defined(_ALLBSD_SOURCE)

 #if defined(__GNUC__) && (__GNUC__ > 4) || (__GNUC__ == 4) && (__GNUC_MINOR__ > 2)
   #define JNIEXPORT     __attribute__((visibility("default")))
diff -r e6e7d76b2bd3 src/cpu/x86/vm/stubGenerator_x86_32.cpp
--- a/src/cpu/x86/vm/stubGenerator_x86_32.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/stubGenerator_x86_32.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -47,6 +47,9 @@
 #ifdef TARGET_OS_FAMILY_windows
 # include "thread_windows.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
```

```
+#endif
 #ifdef COMPILER2
 #include "opto/runtime.hpp"
 #endif
diff -r e6e7d76b2bd3 src/cpu/x86/vm/stubGenerator_x86_64.cpp
--- a/src/cpu/x86/vm/stubGenerator_x86_64.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/stubGenerator_x86_64.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -47,6 +47,9 @@
 #ifdef TARGET_OS_FAMILY_windows
 # include "thread_windows.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif
 #ifdef COMPILER2
 #include "opto/runtime.hpp"
 #endif
diff -r e6e7d76b2bd3 src/cpu/x86/vm/stubRoutines_x86_32.cpp
--- a/src/cpu/x86/vm/stubRoutines_x86_32.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/stubRoutines_x86_32.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -35,6 +35,9 @@
 #ifdef TARGET_OS_FAMILY_windows
 # include "thread_windows.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif

 // Implementation of the platform-specific part of StubRoutines - for
 // a description of how to extend it, see the stubRoutines.hpp file.
diff -r e6e7d76b2bd3 src/cpu/x86/vm/stubRoutines_x86_64.cpp
--- a/src/cpu/x86/vm/stubRoutines_x86_64.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/stubRoutines_x86_64.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -35,6 +35,9 @@
 #ifdef TARGET_OS_FAMILY_windows
 # include "thread_windows.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif

 // Implementation of the platform-specific part of StubRoutines - for
 // a description of how to extend it, see the stubRoutines.hpp file.
diff -r e6e7d76b2bd3 src/cpu/x86/vm/vm_version_x86.cpp
--- a/src/cpu/x86/vm/vm_version_x86.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/x86/vm/vm_version_x86.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -37,6 +37,9 @@
 #ifdef TARGET_OS_FAMILY_windows
 # include "os_windows.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "os_bsd.inline.hpp"
+#endif


 int VM_Version::_cpu;
diff -r e6e7d76b2bd3 src/cpu/zero/vm/bytes_zero.hpp
--- a/src/cpu/zero/vm/bytes_zero.hpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/zero/vm/bytes_zero.hpp        Mon Jul 25 17:04:06 2011 -0700
@@ -168,6 +168,9 @@
 #ifdef TARGET_OS_ARCH_linux_zero
 # include "bytes_linux_zero.inline.hpp"
 #endif
+#ifdef TARGET_OS_ARCH_bsd_zero
+# include "bytes_bsd_zero.inline.hpp"
+#endif

 #endif // VM_LITTLE_ENDIAN

diff -r e6e7d76b2bd3 src/cpu/zero/vm/globals_zero.hpp
--- a/src/cpu/zero/vm/globals_zero.hpp        Tue May 24 15:28:35 2011 -0700
```

```
+++ b/src/cpu/zero/vm/globals_zero.hpp        Mon Jul 25 17:04:06 2011 -0700
@@ -52,7 +52,11 @@
 define_pd_global(bool,  RewriteBytecodes,     true);
 define_pd_global(bool,  RewriteFrequentPairs, true);

+#ifdef _ALLBSD_SOURCE
+define_pd_global(bool,  UseMembar,            true);
+#else
 define_pd_global(bool,  UseMembar,            false);
+#endif

 // GC Ergo Flags
 define_pd_global(intx, CMSYoungGenPerWorker, 16*M);  // default max size of CMS young gen, per GC worker thread
diff -r e6e7d76b2bd3 src/cpu/zero/vm/interp_masm_zero.cpp
--- a/src/cpu/zero/vm/interp_masm_zero.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/zero/vm/interp_masm_zero.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -40,5 +40,8 @@
 #ifdef TARGET_OS_FAMILY_linux
 # include "thread_linux.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif

 // This file is intentionally empty
diff -r e6e7d76b2bd3 src/cpu/zero/vm/stubGenerator_zero.cpp
--- a/src/cpu/zero/vm/stubGenerator_zero.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/zero/vm/stubGenerator_zero.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -43,6 +43,9 @@
 #ifdef TARGET_OS_FAMILY_linux
 # include "thread_linux.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif
 #ifdef COMPILER2
 #include "opto/runtime.hpp"
 #endif
diff -r e6e7d76b2bd3 src/cpu/zero/vm/stubRoutines_zero.cpp
--- a/src/cpu/zero/vm/stubRoutines_zero.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/zero/vm/stubRoutines_zero.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -30,3 +30,6 @@
 #ifdef TARGET_OS_FAMILY_linux
 # include "thread_linux.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "thread_bsd.inline.hpp"
+#endif
diff -r e6e7d76b2bd3 src/cpu/zero/vm/vm_version_zero.cpp
--- a/src/cpu/zero/vm/vm_version_zero.cpp        Tue May 24 15:28:35 2011 -0700
+++ b/src/cpu/zero/vm/vm_version_zero.cpp        Mon Jul 25 17:04:06 2011 -0700
@@ -32,5 +32,8 @@
 #ifdef TARGET_OS_FAMILY_linux
 # include "os_linux.inline.hpp"
 #endif
+#ifdef TARGET_OS_FAMILY_bsd
+# include "os_bsd.inline.hpp"
+#endif

 // This file is intentionally empty
```