DragonFlyBSD - Bug #867

kill db_print_backtrace()

11/28/2007 04:36 PM - aoiko

Status:	Closed	**Start date:**	
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

We've got a lot of #ifdefs in the source that just obfuscate the code
and can be avoided by just using backtrace() from kern_debug.c. This
patch removes all occurunces of db_print_backtrace() along with the
#ifdef DDB, where applicable. Also, let me suggest removing
kern_debug.c::backtrace(), renaming db_print_backtrace() to backtrace()
and using the #ifdef there. The way things are now, any developer can
(out of habit) re-add the #ifdef DDB\n db_print_backtrace(); \n#endif
sequence. Putting

#ifdef DDB
void backtrace(void);
#else
static inline void
backtrace(void)
{
/* maybe kprintf() whatever here */
}
#endif

In any case, I don't think backtrace() needs to be that verbose if DDB
is not available.

Finally, I'd like to remove some similar instances for Debugger(). If
there are no objections, I'll submit patches for the above issues in the
next days.

Index: dev/acpica5/Osd/OsdSynch.c
===
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/dev/acpica5/Osd/OsdSynch.c,v
retrieving revision 1.11
diff -u -p -r1.11 OsdSynch.c
--- dev/acpica5/Osd/OsdSynch.c 25 Jan 2007 15:12:06 -0000 1.11
+++ dev/acpica5/Osd/OsdSynch.c 28 Nov 2007 15:42:08 -0000
@@ -356,7 +356,7 @@ AcpiOsDeleteLock (ACPI_SPINLOCK Spin)
}

#ifdef ACPI_DEBUG_LOCKS
-void db_print_backtrace(void);
+void backtrace(void);
#endif
/*
* OS-dependent locking primitives. These routines should be able to be
@@ -379,7 +379,7 @@ AcpiOsAcquireLock (ACPI_SPINLOCK Spin)
kprintf("%p(%s:%d): acpi_spinlock %p already held by %p(%s:%d)\n",
curthread, func, line, Spin, Spin->owner, Spin->func,
Spin->line);
- db_print_backtrace();
+ backtrace();
} else {
Spin->owner = curthread;
Spin->func = func;
@@ -397,7 +397,7 @@ AcpiOsReleaseLock (ACPI_SPINLOCK Spin, U

```
if (Spin->owner != NULL) {
kprintf("%p: acpi_spinlock %p is unexectedly held by %p(%s:%d)\n",
curthread, Spin, Spin->owner, Spin->func, Spin->line);
-     db_print_backtrace();
+     backtrace();
} else
return;
}
Index: kern/kern_shutdown.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/kern/kern_shutdown.c,v
retrieving revision 1.61
diff -u -p -r1.61 kern_shutdown.c
--- kern/kern_shutdown.c 6 Nov 2007 03:49:58 -0000 1.61
+++ kern/kern_shutdown.c 28 Nov 2007 16:05:26 -0000
@@ -103,8 +103,6 @@ int debugger_on_panic = 1;
SYSCTL_INT(_debug, OID_AUTO, debugger_on_panic, CTLFLAG_RW,
&debugger_on_panic, 0, "Run debugger on kernel panic");

-extern void db_print_backtrace(void);
-
#ifdef DDB_TRACE
int trace_on_panic = 1;
#else
@@ -788,9 +786,9 @@ panic(const char *fmt, ...)
kprintf("cpuid = %d\n", mycpu->gd_cpuid);
#endif

-#if defined(DDB)
if (newpanic && trace_on_panic)
-  db_print_backtrace();
+  backtrace();
+#if defined(DDB)
if (debugger_on_panic)
Debugger("panic");
#endif
Index: kern/kern_spinlock.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/kern/kern_spinlock.c,v
retrieving revision 1.11
diff -u -p -r1.11 kern_spinlock.c
--- kern/kern_spinlock.c 2 Jul 2007 16:51:58 -0000 1.11
+++ kern/kern_spinlock.c 28 Nov 2007 15:44:30 -0000
@@ -273,16 +273,16 @@ exponential_backoff(struct exponential_b
kprintf("spin_lock: %p, indefinite wait!\n", bo->mtx);
if (panicstr)
return (TRUE);
-#if defined(INVARIANTS) && defined(DDB)
+#if defined(INVARIANTS)
if (spin_lock_test_mode) {
-   db_print_backtrace();
+   backtrace();
return (TRUE);
}
#endif
++bo->nsec;
-#if defined(INVARIANTS) && defined(DDB)
+#if defined(INVARIANTS)
if (bo->nsec == 11)
-   db_print_backtrace();
+   backtrace();
#endif
if (bo->nsec == 60)
panic("spin_lock: %p, indefinite wait!\n", bo->mtx);
Index: kern/kern_timeout.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/kern/kern_timeout.c,v
```

```
retrieving revision 1.27
diff -u -p -r1.27 kern_timeout.c
--- kern/kern_timeout.c 14 Nov 2007 18:27:52 -0000 1.27
+++ kern/kern_timeout.c 28 Nov 2007 15:45:30 -0000
@@ -360,9 +360,7 @@ callout_reset(struct callout *c, int to_
kprintf(
"callout_reset(%p) from %p: callout was not initialized\n",
c, ((int **)&c)[-1]);
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
#endif
gd = mycpu;
@@ -415,9 +413,7 @@ callout_stop(struct callout *c)
kprintf(
"callout_stop(%p) from %p: callout was not initialized\n",
c, ((int **)&c)[-1]);
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
#endif
crit_enter_gd(gd);
Index: kern/lwkt_thread.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/kern/lwkt_thread.c,v
retrieving revision 1.110
diff -u -p -r1.110 lwkt_thread.c
--- kern/lwkt_thread.c 27 Sep 2007 18:27:54 -0000 1.110
+++ kern/lwkt_thread.c 28 Nov 2007 15:46:10 -0000
@@ -500,9 +500,7 @@ lwkt_switch(void)
td->td_flags |= TDF_PANICWARN;
kprintf("Warning: thread switch from interrupt or IPI, "
"thread %p (%s)\n", td, td->td_comm);
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
lwkt_switch();
gd->gd_intr_nesting_level = savegdnest;
Index: kern/uipc_mbuf.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/kern/uipc_mbuf.c,v
retrieving revision 1.65
diff -u -p -r1.65 uipc_mbuf.c
--- kern/uipc_mbuf.c 12 Aug 2007 01:46:26 -0000 1.65
+++ kern/uipc_mbuf.c 28 Nov 2007 15:47:25 -0000
@@ -850,8 +850,6 @@ m_mclfree(void *arg)
objcache_put(mclmeta_cache, mcl);
}

-extern void db_print_backtrace(void);
-
/*
* Free a single mbuf and any associated external storage.  The successor,
* if any, is returned.
@@ -881,14 +879,12 @@ m_free(struct mbuf *m)
KKASSERT(m->m_nextpkt == NULL);
#else
if (m->m_nextpkt != NULL) {
-#ifdef DDB
static int afewtimes = 10;
```

```
if (afewtimes-- > 0) {
kprintf("mfree: m->m_nextpkt != NULL\n");
-   db_print_backtrace();
+   backtrace();
}
-#endif
m->m_nextpkt = NULL;
}
#endif
Index: net/route.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/net/route.c,v
retrieving revision 1.32
diff -u -p -r1.32 route.c
--- net/route.c 9 Aug 2007 01:10:05 -0000 1.32
+++ net/route.c 28 Nov 2007 15:48:08 -0000
@@ -1302,7 +1302,7 @@ rt_addrinfo_print(int cmd, struct rt_add

#ifdef ROUTE_DEBUG
if (cmd == RTM_DELETE && route_debug > 1)
-  db_print_backtrace();
+  backtrace();
#endif

switch(cmd) {
Index: platform/pc32/i386/pmap.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/platform/pc32/i386/pmap.c,v
retrieving revision 1.81
diff -u -p -r1.81 pmap.c
--- platform/pc32/i386/pmap.c 15 Aug 2007 03:15:07 -0000 1.81
+++ platform/pc32/i386/pmap.c 28 Nov 2007 15:48:58 -0000
@@ -1930,15 +1930,11 @@ pmap_enter(pmap_t pmap, vm_offset_t va,
#endif
if (va < UPT_MAX_ADDRESS && pmap == &kernel_pmap) {
kprintf("Warning: pmap_enter called on UVA with kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
if (va >= UPT_MAX_ADDRESS && pmap != &kernel_pmap) {
kprintf("Warning: pmap_enter called on KVA without kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}

mpte = NULL;
@@ -2089,15 +2085,11 @@ pmap_enter_quick(pmap_t pmap, vm_offset_

if (va < UPT_MAX_ADDRESS && pmap == &kernel_pmap) {
kprintf("Warning: pmap_enter_quick called on UVA with kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
if (va >= UPT_MAX_ADDRESS && pmap != &kernel_pmap) {
kprintf("Warning: pmap_enter_quick called on KVA without kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
```

```
/*
Index: vm/vm_vmspace.c
===================================================================
RCS file: /home/aggelos/imports/vcs/dcvs/src/sys/vm/vm_vmspace.c,v
retrieving revision 1.14
diff -u -p -r1.14 vm_vmspace.c
--- vm/vm_vmspace.c 15 Aug 2007 03:15:07 -0000 1.14
+++ vm/vm_vmspace.c 28 Nov 2007 15:49:32 -0000
@@ -481,9 +481,7 @@ vkernel_lwp_exit(struct lwp *lp)
if ((ve = vklp->ve) != NULL) {
kprintf("Warning, pid %d killed with "
"active VC!\n", lp->lwp_proc->p_pid);
-#ifdef DDB
-   db_print_backtrace();
-#endif
+   backtrace();
pmap_setlwpvm(lp, lp->lwp_proc->p_vmspace);
vklp->ve = NULL;
KKASSERT(ve->refs > 0);
```

History

#1 - 11/29/2007 01:17 AM - qhwt+dfly

You can drop this prototype declaration and the surrounding #ifdef too.

Cheers.

#2 - 11/29/2007 12:24 PM - aoiko

Oops. Updated patch follows.

```
Index: dev/acpica5/Osd/OsdSynch.c
===================================================================
retrieving revision 1.11
diff -u -p -u -r1.11 OsdSynch.c
--- dev/acpica5/Osd/OsdSynch.c
+++ dev/acpica5/Osd/OsdSynch.c
@@ -355,9 +355,6 @@ AcpiOsDeleteLock (ACPI_SPINLOCK Spin)
kfree(Spin, M_ACPISEM);
}

-#ifdef ACPI_DEBUG_LOCKS
-void db_print_backtrace(void);
-#endif
/*
* OS-dependent locking primitives.  These routines should be able to be
* called from an interrupt-handler or cpu_idle thread.
@@ -379,7 +376,7 @@ AcpiOsAcquireLock (ACPI_SPINLOCK Spin)
kprintf("%p(%s:%d): acpi_spinlock %p already held by %p(%s:%d)\n",
curthread, func, line, Spin, Spin->owner, Spin->func,
Spin->line);
- db_print_backtrace();
+ backtrace();
} else {
Spin->owner = curthread;
Spin->func = func;
@@ -397,7 +394,7 @@ AcpiOsReleaseLock (ACPI_SPINLOCK Spin, U
if (Spin->owner != NULL) {
kprintf("%p: acpi_spinlock %p is unexectedly held by %p(%s:%d)\n",
curthread, Spin, Spin->owner, Spin->func, Spin->line);
-    db_print_backtrace();
+    backtrace();
} else
return;
}
Index: kern/kern_shutdown.c
===================================================================
retrieving revision 1.61
diff -u -p -u -r1.61 kern_shutdown.c
```

```
--- kern/kern_shutdown.c
+++ kern/kern_shutdown.c
@@ -103,8 +103,6 @@ int debugger_on_panic = 1;
SYSCTL_INT(_debug, OID_AUTO, debugger_on_panic, CTLFLAG_RW,
&debugger_on_panic, 0, "Run debugger on kernel panic");

-extern void db_print_backtrace(void);
-
#ifdef DDB_TRACE
int trace_on_panic = 1;
#else
@@ -788,9 +786,9 @@ panic(const char *fmt, ...)
kprintf("cpuid = %d\n", mycpu->gd_cpuid);
#endif

-#if defined(DDB)
if (newpanic && trace_on_panic)
-  db_print_backtrace();
+  backtrace();
+#if defined(DDB)
if (debugger_on_panic)
Debugger("panic");
#endif
Index: kern/kern_spinlock.c
===================================================================
retrieving revision 1.11
diff -u -p -u -r1.11 kern_spinlock.c
--- kern/kern_spinlock.c
+++ kern/kern_spinlock.c
@@ -273,16 +273,16 @@ exponential_backoff(struct exponential_b
kprintf("spin_lock: %p, indefinite wait!\n", bo->mtx);
if (panicstr)
return (TRUE);
-#if defined(INVARIANTS) && defined(DDB)
+#if defined(INVARIANTS)
if (spin_lock_test_mode) {
-   db_print_backtrace();
+   backtrace();
return (TRUE);
}
#endif
++bo->nsec;
-#if defined(INVARIANTS) && defined(DDB)
+#if defined(INVARIANTS)
if (bo->nsec == 11)
-   db_print_backtrace();
+   backtrace();
#endif
if (bo->nsec == 60)
panic("spin_lock: %p, indefinite wait!\n", bo->mtx);
Index: kern/kern_timeout.c
===================================================================
retrieving revision 1.27
diff -u -p -u -r1.27 kern_timeout.c
--- kern/kern_timeout.c
+++ kern/kern_timeout.c
@@ -360,9 +360,7 @@ callout_reset(struct callout *c, int to_
kprintf(
"callout_reset(%p) from %p: callout was not initialized\n",
c, ((int **)&c)[-1]);
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
#endif
gd = mycpu;
@@ -415,9 +413,7 @@ callout_stop(struct callout *c)
kprintf(
"callout_stop(%p) from %p: callout was not initialized\n",
c, ((int **)&c)[-1]);
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
```

```
}
#endif
crit_enter_gd(gd);
Index: kern/lwkt_thread.c
===================================================================
retrieving revision 1.110
diff -u -p -u -r1.110 lwkt_thread.c
--- kern/lwkt_thread.c
+++ kern/lwkt_thread.c
@@ -500,9 +500,7 @@ lwkt_switch(void)
td->td_flags |= TDF_PANICWARN;
kprintf("Warning: thread switch from interrupt or IPI, "
"thread %p (%s)\n", td, td->td_comm);
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
lwkt_switch();
gd->gd_intr_nesting_level = savegdnest;
Index: kern/uipc_mbuf.c
===================================================================
retrieving revision 1.65
diff -u -p -u -r1.65 uipc_mbuf.c
--- kern/uipc_mbuf.c
+++ kern/uipc_mbuf.c
@@ -850,8 +850,6 @@ m_mclfree(void *arg)
objcache_put(mclmeta_cache, mcl);
}

-extern void db_print_backtrace(void);
-
/*
* Free a single mbuf and any associated external storage.  The successor,
* if any, is returned.
@@ -881,14 +879,12 @@ m_free(struct mbuf *m)
KKASSERT(m->m_nextpkt == NULL);
#else
if (m->m_nextpkt != NULL) {
-#ifdef DDB
static int afewtimes = 10;

if (afewtimes-- > 0) {
kprintf("mfree: m->m_nextpkt != NULL\n");
-   db_print_backtrace();
+   backtrace();
}
-#endif
m->m_nextpkt = NULL;
}
#endif
Index: net/route.c
===================================================================
retrieving revision 1.32
diff -u -p -u -r1.32 route.c
--- net/route.c
+++ net/route.c
@@ -1302,7 +1302,7 @@ rt_addrinfo_print(int cmd, struct rt_add

#ifdef ROUTE_DEBUG
if (cmd == RTM_DELETE && route_debug > 1)
-  db_print_backtrace();
+  backtrace();
#endif

switch(cmd) {
Index: platform/pc32/i386/pmap.c
===================================================================
retrieving revision 1.81
diff -u -p -u -r1.81 pmap.c
--- platform/pc32/i386/pmap.c
+++ platform/pc32/i386/pmap.c
@@ -1930,15 +1930,11 @@ pmap_enter(pmap_t pmap, vm_offset_t va,
#endif
if (va < UPT_MAX_ADDRESS && pmap == &kernel_pmap) {
```

```
kprintf("Warning: pmap_enter called on UVA with kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
if (va >= UPT_MAX_ADDRESS && pmap != &kernel_pmap) {
kprintf("Warning: pmap_enter called on KVA without kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}

mpte = NULL;
@@ -2089,15 +2085,11 @@ pmap_enter_quick(pmap_t pmap, vm_offset_

if (va < UPT_MAX_ADDRESS && pmap == &kernel_pmap) {
kprintf("Warning: pmap_enter_quick called on UVA with kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}
if (va >= UPT_MAX_ADDRESS && pmap != &kernel_pmap) {
kprintf("Warning: pmap_enter_quick called on KVA without kernel_pmap\n");
-#ifdef DDB
-  db_print_backtrace();
-#endif
+  backtrace();
}

/*
Index: vm/vm_vmspace.c
===================================================================
retrieving revision 1.14
diff -u -p -u -r1.14 vm_vmspace.c
--- vm/vm_vmspace.c
+++ vm/vm_vmspace.c
@@ -481,9 +481,7 @@ vkernel_lwp_exit(struct lwp *lp)
if ((ve = vklp->ve) != NULL) {
kprintf("Warning, pid %d killed with "
"active VC!\n", lp->lwp_proc->p_pid);
-#ifdef DDB
-   db_print_backtrace();
-#endif
+   backtrace();
pmap_setlwpvm(lp, lp->lwp_proc->p_vmspace);
vklp->ve = NULL;
KKASSERT(ve->refs > 0);
```

#3 - 01/20/2009 11:18 AM - hasso

What's the opinion regarding this? It should be either committed or closed.

#4 - 01/20/2009 11:56 AM - sepherosa

On Tue, Jan 20, 2009 at 7:18 PM, Hasso Tepper (via DragonFly issue
tracker) <sinknull@crater.dragonflybsd.org> wrote:
>
> Hasso Tepper <hasso@estpak.ee> added the comment:
>
> What's the opinion regarding this? It should be either committed or closed.

Looks good to me. However, should ddb/ddb.h uninclude in these files
after this patch?

Best Regards,
sephe

#5 - 02/05/2009 06:40 PM - aoiko

This report today:

<jwh> hm, guys
<jwh> linking kernel.nodebug
<jwh> route.o: In function `rtfree_remote':
<jwh> route.c:(.text+0xfb): undefined reference to `db_print_backtrace'
<jwh> ip_output.o: In function `ip_output':
<jwh> ip_output.c:(.text+0x5c2): undefined reference to `db_print_backtrace'
<jwh> building vkernel

prompted me to produce a new patch, which indeed removes some #ifdef DDB
and #include <ddb/ddb.h> lines. Compile tested w/ LINT and my regular
kernel config without DDB.

Opinions? Otherwise in it goes, should be trivial enough for the soft
freeze :)

Sorry this is not inline, icedove sucks quite a bit for serious email
usage...

Aggelos

#6 - 02/05/2009 10:38 PM - aoiko

Committed, renamed backtrace() to print_backtrace() to be future-compatible as requested by corecode.

Files

db_print_backtrace_removal.patch	10.9 KB	02/06/2009	aoiko