

---

Subject: [PATCH 1/1] tests: add simple forker test  
Posted by [serue](#) on Tue, 24 Jun 2008 17:43:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Another braindead but useful test. The point here is to start testing cryo on multiple tasks.

If you run  
./tests/fork 0  
then restart works fine. If you run  
./tests/fork 1 # (or anything greater than 1)  
then there is a segfault on restart. It looks a lot like the problem which Benjamin solved by re-enabling the NPTL code, but I haven't quite tracked it down yet.

-serge

>From 8dd009389ada5106ad8c5d8b6e565ec75f13177a Mon Sep 17 00:00:00 2001  
From: Serge E. Hallyn <serue@us.ibm.com>  
Date: Tue, 24 Jun 2008 10:12:34 -0400  
Subject: [PATCH 1/1] tests: add simple forker test

Add a test which forks atoi(argv[1]) times, then sleeps 1 min.

Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>

---  
tests/Makefile | 2 +-  
tests/fork.c | 53 +++  
2 files changed, 54 insertions(+), 1 deletions(-)  
create mode 100644 tests/fork.c

```
diff --git a/tests/Makefile b/tests/Makefile
index 925b7af..f8233a2 100644
--- a/tests/Makefile
+++ b/tests/Makefile
@@ -1,4 +1,4 @@
-TARGETS = sleep mksysvipc pause_asm pipe sockudp
+TARGETS = sleep mksysvipc pause_asm pipe sockudp fork
```

CFLAGS = -static

```
diff --git a/tests/fork.c b/tests/fork.c
new file mode 100644
index 0000000..955d475
--- /dev/null
+++ b/tests/fork.c
@@ -0,0 +1,53 @@
+#include <stdio.h>
```

```

#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <signal.h>
#include <string.h>
+
+void sighandler(int sig)
+{
+ printf("%d: got signal %d\n", getpid(), sig);
+}
+
+void do_sleep(int whoami)
+{
+ int i;
+ for (i=0; i<30; i++)
+ sleep(2);
+ printf("%d: exiting\n", whoami);
+ exit(0);
+}
+
+int main(int argc, char *argv[])
+{
+ int nt, i;
+ int *pids;
+ struct sigaction s;
+
+ if (argc<2) {
+ exit(1);
+ }
+
+ memset(&s, 0, sizeof(s));
+ s.sa_handler = sighandler;
+ sigaction(SIGUSR1, &s, NULL);
+ nt = atoi(argv[1]);
+ pids = malloc(nt);
+ if (!pids)
+ exit(2);
+ for (i=0; i<nt; i++) {
+ pids[i] = fork();
+ if (pids[i]<0) {
+ perror("fork");
+ exit(3);
+ } else if (pids[i] == 0)
+ do_sleep(i);
+ printf("forked %d\n", i);
+ }
+ for (i=0; i<30; i++)
+ sleep(2);

```

```
+ free(pids);  
+ printf("main: exiting\n");  
+ exit(0);  
+}
```

--

1.5.4.1

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---