Subject: [PATCH 2/4] add the init-logger binary
Posted by dietmar on Tue, 20 Sep 2011 05:17:16 GMT
View Forum Message <> Reply to Message

From: Dietmar Maurer <dietmar@proxmox.com>

We use diet to create small binaries.

Signed-off-by: Dietmar Maurer <dietmar@proxmox.com>

```
---
 scripts/Makefile.am    |    4 +
 scripts/init-logger.c |  163 +++++++++++++++++++++++++++++++++++++++++++++++++
 2 files changed, 167 insertions(+), 0 deletions(-)
 create mode 100644 scripts/init-logger.c

diff --git a/scripts/Makefile.am b/scripts/Makefile.am
index 5b28c6e..3d25b6a 100644
--- a/scripts/Makefile.am
+++ b/scripts/Makefile.am
@@ -17,7 +17,11 @@

 include $(top_srcdir)/pathsubst.am

+init-logger: init-logger.c
+ diet -Os gcc -static -s -o init-logger init-logger.c
+
 vzlib_SCRIPTS = \
+	init-logger \
  vps-create \
  vps-functions \
  vps-net_add \
diff --git a/scripts/init-logger.c b/scripts/init-logger.c
new file mode 100644
index 0000000..1153b1b
--- /dev/null
+++ b/scripts/init-logger.c
@@ -0,0 +1,163 @@
+/*
+
+    Copyright (C) 2008 Proxmox Server Solutions GmbH
+
+    This program is free software; you can redistribute it and/or modify
+    it under the terms of the GNU General Public License as published by
+    the Free Software Foundation; version 2 dated June, 1991.
+
+    This program is distributed in the hope that it will be useful,
+    but WITHOUT ANY WARRANTY; without even the implied warranty of
+    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
```

```
+     GNU General Public License for more details.
+
+     You should have received a copy of the GNU General Public License
+     along with this program; if not, write to the Free Software Foundation,
+     Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
+     You should have received a copy of the GNU General Public License
+
+     Author: Dietmar Maurer <dietmar@proxmox.com>
+
+     Compile statically using dietlibc:
+     diet -Os gcc -static -s -o init-logger.ARCH init-logger.c
+*/
+
+#include <sys/types.h>
+#include <sys/wait.h>
+#include <stdlib.h>
+#include <unistd.h>
+#include <errno.h>
+#include <stdio.h>
+#include <signal.h>
+#include <sys/select.h>
+
+/* Set a signal handler */
+static void
+setsig(struct sigaction *sa, int sig,
+     void (*fun)(int), int flags)
+{
+ sa->sa_handler = fun;
+ sa->sa_flags = flags;
+ sigemptyset(&sa->sa_mask);
+ sigaction(sig, sa, NULL);
+}
+
+static int terminate = 0;
+
+void
+term_handler()
+{
+ terminate = 1;
+}
+
+ssize_t
+safe_read (int fd, char *buf, size_t count)
+{
+ ssize_t n;
+
+ do {
+   n = read (fd, buf, count);
```

```
+ } while (n < 0 && errno == EINTR);
+
+ return n;
+}
+
+ssize_t
+safe_write (int fd, char *buf, size_t count)
+{
+ ssize_t n;
+
+ do {
+  n = write (fd, buf, count);
+ } while (n < 0 && errno == EINTR);
+
+ return n;
+}
+
+int
+full_write(int fd, char *buf, size_t len)
+{
+ size_t n;
+ size_t total;
+
+ total = 0;
+
+ while (len > 0) {
+         n = safe_write(fd, buf, len);
+
+  if (n < 0)
+           break;
+
+  buf += n;
+  total += n;
+  len -= n;
+ }
+
+ return total;
+}
+
+static void
+simple_cat (void)
+{
+        int bufsize = 256;
+ char buf[bufsize];
+ size_t n_read;
+ int noop_count = 0;
+
+ fd_set rfds;
```

```
+ struct timeval tv;
+ int retval;
+
+ FD_ZERO(&rfds);
+ FD_SET(STDIN_FILENO, &rfds);
+
+ tv.tv_sec = 1;
+ tv.tv_usec = 0;
+
+ while ((retval = select(STDIN_FILENO + 1, &rfds, NULL, NULL, &tv)) >= 0 ||
+        (errno == EINTR)) {
+
+  tv.tv_sec = 1;
+  tv.tv_usec = 0;
+
+  FD_ZERO(&rfds);
+  FD_SET(STDIN_FILENO, &rfds);
+
+  if (retval == -1 && errno == EINTR)
+      continue;
+
+  if (retval) {
+   n_read = safe_read (STDIN_FILENO, buf, bufsize);
+   if (n_read ==  ((size_t) -1))
+    return;
+
+   noop_count = 0;
+
+   if (full_write (STDOUT_FILENO, buf, n_read) != n_read)
+    return;
+  } else {
+   if (terminate)
+    noop_count++;
+  }
+
+  if (noop_count >= 2)
+   return;
+ }
+}
+
+int
+main(int argc, char * argv[])
+{
+ struct sigaction sa;
+
+ setsig(&sa, SIGTERM, term_handler, SA_RESTART);
+ setsig(&sa, SIGINT, term_handler, SA_RESTART);
+
```

```
+ printf ("starting init logger\n");
+
+ simple_cat();
+
+ printf ("\ninit logger finished\n");
+
+ exit (0);
+}
--
1.5.6.5
```