```
**********************************************************
   86454 Mon Jan 21 14:34:00 2013
new/usr/src/cmd/cron/cron.c
374 cron should send more useful mail
Reviewed by: Richard Lowe <richlowe@richlowe.net>
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License (the "License").
 * You may not use this file except in compliance with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 *
 * Copyright 2013 Joshua M. Clulow <josh@sysmgr.org>
 */

/*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
/*        All Rights Reserved   */

/*      Copyright (c) 1987, 1988 Microsoft Corporation  */
/*        All Rights Reserved   */

#ifdef lint
/* make lint happy */
#define __EXTENSIONS__
#endif

#include <sys/contract/process.h>
#include <sys/ctfs.h>
#include <sys/param.h>
#include <sys/resource.h>
#include <sys/stat.h>
#include <sys/task.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/utsname.h>
#include <sys/wait.h>

#include <security/pam_appl.h>

#include <alloca.h>
#include <ctype.h>
#include <deflt.h>
#include <dirent.h>
#include <errno.h>
#include <fcntl.h>
#include <grp.h>
#include <libcontract.h>
#include <libcontract_priv.h>
```



```
#include <limits.h>
#include <locale.h>
#include <poll.h>
#include <project.h>
#include <pwd.h>
#include <signal.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stropts.h>
#include <time.h>
#include <unistd.h>
#include <libzoneinfo.h>

#include "cron.h"

/*
 * #define      DEBUG
 */

#define MAIL            "/usr/bin/mail" /* mail program to use */
#define CONSOLE         "/dev/console"  /* where messages go when cron dies */

#define TMPINFILE       "/tmp/crinXXXXXX"  /* file to put stdin in for cmd  */
#define TMPDIR          "/tmp"
#define PFX             "crout"
#define TMPOUTFILE      "/tmp/croutXXXXXX" /* file to place stdout, stderr */

#define INMODE          00400           /* mode for stdin file  */
#define OUTMODE         00600           /* mode for stdout file */
#define ISUID           S_ISUID         /* mode for verifing at jobs */

#define INFINITY        2147483647L     /* upper bound on time  */
#define CUSHION         180L
#define ZOMB            100             /* proc slot used for mailing output */

#define JOBF            'j'
#define NICEF           'n'
#define USERF           'u'
#define WAITF           'w'

#define BCHAR           '>'
#define ECHAR           '<'

#define DEFAULT         0
#define LOAD            1
#define QBUFSIZ         80

/* Defined actions for crabort() routine */
#define NO_ACTION       000
#define REMOVE_FIFO     001
#define CONSOLE_MSG     002

#define BADCD           "can't change directory to the crontab directory."
#define NOREADDIR       "can't read the crontab directory."

#define BADJOBOPEN      "unable to read your at job."
#define BADSHELL        "because your login shell \
isn't /usr/bin/sh, you can't use cron."

#define BADSTAT         "can't access your crontab or at-job file. Resubmit it."
#define BADPROJID       "can't set project id for your job."
#define CANTCDHOME      "can't change directory to %s.\
\nYour commands will not be executed."
#define CANTEXECSH      "unable to exec the shell, %s, for one of your \
```

```
 127 commands."
 128 #define CANT_STR_LEN (sizeof (CANTEXECSH) > sizeof (CANTCDHOME) ? \
 129         sizeof (CANTEXECSH) : sizeof (CANTCDHOME))
 130 #define NOREAD          "can't read your crontab file.  Resubmit it."
 131 #define BADTYPE         "crontab or at-job file is not a regular file.\n"
 132 #define NOSTDIN         "unable to create a standard input file for \
 133 one of your crontab commands. \
 134 \nThat command was not executed."

 136 #define NOTALLOWED      "you are not authorized to use cron.  Sorry."
 137 #define STDERRMSG       "\n\n*****************************************\
 138 *****\nCron: The previous message is the \
 139 standard output and standard error \
 140 \nof one of your cron commands.\n"

 142 #define STDOUTERR       "one of your commands generated output or errors, \
 143 but cron was unable to mail you this output.\
 144 \nRemember to redirect standard output and standard \
 145 error for each of your commands."

 147 #define CLOCK_DRIFT     "clock time drifted backwards after event!\n"
 148 #define PIDERR          "unexpected pid returned %d (ignored)"
 149 #define CRONTABERR      "Subject: Your crontab file has an error in it\n\n"
 148 #define CRONOUT         "Subject: Output from \"cron\" command\n\n"
 150 #define MALLOCERR       "out of space, cannot create new string\n"

 152 #define DIDFORK didfork
 153 #define NOFORK !didfork

 155 #define MAILBUFLEN      (8*1024)
 156 #define LINELIMIT       80
 157 #define MAILBINITFREE   (MAILBUFLEN - (sizeof (cte_intro) - 1) \
 158             - (sizeof (cte_trail1) - 1) - (sizeof (cte_trail2) - 1) - 1)

 160 #define ERR_CRONTABENT  0       /* error in crontab file entry */
 161 #define ERR_UNIXERR     1       /* error in some system call */
 162 #define ERR_CANTEXECCRON 2      /* error setting up "cron" job environment */
 163 #define ERR_CANTEXECAT  3       /* error setting up "at" job environment */
 164 #define ERR_NOTREG      4       /* error not a regular file */

 166 #define PROJECT         "project="

 168 #define MAX_LOST_CONTRACTS      2048    /* reset if this many failed abandons */

 170 #define FORMAT  "%a %b %e %H:%M:%S %Y"
 171 static char     timebuf[80];

 173 static struct message msgbuf;

 175 struct shared {
 176         int count;                      /* usage count */
 177         void (*free)(void *obj);        /* routine that will free obj */
 178         void *obj;                      /* object */
 179 };
```

*_unchanged_portion_omitted_*

```
2700 /*
2701  * Mail stdout and stderr of a job to user. Get uid for real user and become
2702  * that person. We do this so that mail won't come from root since this
2703  * could be a security hole. If failure, quit - don't send mail as root.
2704  */
2705 static void
2706 mail_result(struct usr *p, struct runinfo *pr, size_t filesize)
2707 {
2708         struct  passwd  *ruser_ids;
2709         FILE    *mailpipe;
```



```
2710         FILE    *st;
2711         struct utsname  name;
2712         int     nbytes;
2713         char    iobuf[BUFSIZ];
2714         char    *cmd;
2715         char    *lowname = (pr->jobtype == CRONEVENT ? "cron" : "at");

2717         (void) uname(&name);
2718         if ((ruser_ids = getpwnam(p->name)) == NULL)
2719                 exit(0);
2720         (void) setuid(ruser_ids->pw_uid);

2722         cmd = xmalloc(strlen(MAIL) + strlen(p->name)+2);
2723         (void) sprintf(cmd, "%s %s", MAIL, p->name);
2724         mailpipe = popen(cmd, "w");
2725         free(cmd);
2726         if (mailpipe == NULL)
2727                 exit(127);
2728         (void) fprintf(mailpipe, "To: %s\n", p->name);
2729         (void) fprintf(mailpipe, "Subject: %s <%s@%s> %s\n",
2730             (pr->jobtype == CRONEVENT ? "Cron" : "At"),
2731             p->name, name.nodename, pr->jobname);

2733         /*
2734          * RFC3834 (Section 5) defines the Auto-Submitted header to prevent
2735          * vacation replies, et al, from being sent in response to
2736          * machine-generated mail.
2737          */
2738         (void) fprintf(mailpipe, "Auto-Submitted: auto-generated\n");

2740         /*
2741          * Additional headers for mail filtering and diagnostics:
2742          */
2743         (void) fprintf(mailpipe, "X-Mailer: cron (%s %s)\n", name.sysname,
2744             name.release);
2745         (void) fprintf(mailpipe, "X-Cron-User: %s\n", p->name);
2746         (void) fprintf(mailpipe, "X-Cron-Host: %s\n", name.nodename);
2747         (void) fprintf(mailpipe, "X-Cron-Job-Name: %s\n", pr->jobname);
2748         (void) fprintf(mailpipe, "X-Cron-Job-Type: %s\n", lowname);

2750         /*
2751          * Message Body:
2752          *
2753          * (Temporary file is fopen'ed with "r", secure open.)
2754          */
2755         (void) fprintf(mailpipe, "\n");
2727         if (pr->jobtype == CRONEVENT) {
2728                 (void) fprintf(mailpipe, CRONOUT);
2729                 (void) fprintf(mailpipe, "Your \"cron\" job on %s\n",
2730                     name.nodename);
2731                 if (pr->jobname != NULL) {
2732                         (void) fprintf(mailpipe, "%s\n\n", pr->jobname);
2733                 }
2734         } else {
2735                 (void) fprintf(mailpipe, "Subject: Output from \"at\" job\n\n");
2736                 (void) fprintf(mailpipe, "Your \"at\" job on %s\n",
2737                     name.nodename);
2738                 if (pr->jobname != NULL) {
2739                         (void) fprintf(mailpipe, "\"%s\"\n\n", pr->jobname);
2740                 }
2741         }
2742         /* Tmp. file is fopen'ed w/ "r",  secure open */
2756         if (filesize > 0 &&
2757             (st = fopen(pr->outfile, "r")) != NULL) {
2745                 (void) fprintf(mailpipe,
2746                     "produced the following output:\n\n");
```

```
2758                 while ((nbytes = fread(iobuf, sizeof (char), BUFSIZ, st)) != 0)
2759                         (void) fwrite(iobuf, sizeof (char), nbytes, mailpipe);
2760                 (void) fclose(st);
2761         } else {
2762                 (void) fprintf(mailpipe, "Job completed with no output.\n");
2751                 (void) fprintf(mailpipe, "completed.\n");
2763         }
2764         (void) pclose(mailpipe);
2765         exit(0);
2766 }
```

_____***unchanged_portion_omitted_***