POSIX/GNU

Links

1. General
    - Wikipedia: POSIX
    - IEEE and The Open Group: The Open Group Base Specifications Issue 7
2. Special Topics
    - Free Software Foundation: The GNU C Library Reference Manual - Generating Unpredictable Bytes (`getrandom`)
    - Free Software Foundation: The GNU C Library Reference Manual - Erasing Sensitive Data (`explicit_bzero`)
    - Free Software Foundation: The GNU C Library Reference Manual - Implementing a Job Control Shell
    - Thomas Orozco: A Deep Dive into the SIGTTIN / SIGTTOU Terminal Access Control Mechanism in Linux
3. Inlined (see below)
    - Execution Time & Timestamps (gettimeofday)
    - Name of Current Process (getpid)
    - Change *NIX User Password

Execution Time & Timestamps (gettimeofday)

```
#include <sys/time.h>

struct timeval t;
uint64_t start;

if (gettimeofday(&t, NULL) == 0)
{
    start = 1000000ULL * t.tv_sec + t.tv_usec;
} // if

:
:

if (gettimeofday(&t, NULL) == 0)
{
    printf("%llu us", (1000000ULL * t.tv_sec + t.tv_usec) - start);
} // if
```

The resolution of function `gettimeofday()` depends on the platform, but mostly all *Linux* like systems implement it using high resolution timers.

Name of Current Process (getpid)

The following code fragment determines the executable name (possibly incl. path) of current process in variable exe.

```
#include <unistd.h>
#include <stdio.h>

char exe[256]; /* to be adopted, possibly by dynamic alloc */
char path[32]; /* path of 'exe' link in proc filesystem */
ssize_t len;

snprintf(path, sizeof(path), "/proc/%d/exe", (int) getpid()); /* not
portable */
buf[sizeof(path) - 1] = '\0';
len = readlink(path, exe, sizeof(exe));

if (len != -1)
    exe[len] = '\0';
else
    ; /* implement error handling */
```

The implementation works under *Linux* and some other *nix derivates, but not for all systems (you might use `#ifdef __linux__` if this method is appropriate).

See man page `proc(5)` for details on *proc* filesystem under *Linux*.

Change *NIX User Password

```
int change_passwd ( const char* user, const char* oldpw, const char* newpw )
{
    int fds[2];
    int xstatus = EXIT_FAILURE; /* exit status */

    if ( pipe ( fds ) == 0 )
    {
        pid_t pid = fork ( );

        if ( pid >= 0 )
        {
            if ( pid == 0 ) /* in child process ? */
            {
                close ( fds[1] ); /* close write end */

                if ( dup2 ( fds[0], STDIN_FILENO ) >= 0 )
```

```
                {
                    execl ( "/usr/bin/passwd", "passwd", user, (char *) NULL
);
                }

                exit ( xstatus );
            }
            else
            {
                int status; /* child status */
                ssize_t len = strlen ( newpw );

                close ( fds[0] );
                xstatus = EXIT_SUCCESS;

                if ( oldpw != NULL ) /* normal user (not root) ? */
                {
                    if ( ( write ( fds[1], oldpw, strlen ( oldpw ) ) < 0 )
||
                         ( write ( fds[1], "\n", 1 ) != 1 ) )
                    {
                        xstatus = EXIT_FAILURE;
                    }
                }

                if ( xstatus == EXIT_SUCCESS )
                {
                    if ( ( write ( fds[1], newpw, len ) != len ) ||
                         ( write ( fds[1], "\n", 1 ) != 1 ) ||
                         ( write ( fds[1], newpw, len ) != len ) ||
                         ( write ( fds[1], "\n", 1 ) != 1 ) )
                    {
                        xstatus = EXIT_FAILURE;
                    }
                }

                close ( fds[1] );

                if ( waitpid ( pid, &status, 0 ) >= 0 )
                {
                    if ( WIFEXITED ( status ) && ( xstatus == EXIT_SUCCESS )
)
                    {
                        xstatus = WEXITSTATUS ( status );
                    }
                }
                else
                {
                    kill ( pid, SIGKILL ); /* sanity */
                }
            }
```

```
        }
    }

    return xstatus;
}
```

From:
https://wiki.rho62.de/ - **rho62 Wiki**

Permanent link:
https://wiki.rho62.de/doku.php?id=programming:posix

Last update: **2024/02/12 11:32**