



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'setpgid.3p' command***

### ***\$ man setpgid.3p***

SETPGID(3P)            POSIX Programmer's Manual            SETPGID(3P)

#### PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

#### NAME

setpgid ? set process group ID for job control

#### SYNOPSIS

```
#include <unistd.h>

int setpgid(pid_t pid, pid_t pgid);
```

#### DESCRIPTION

The setpgid() function shall either join an existing process group or create a new process group within the session of the calling process.

The process group ID of a session leader shall not change.

Upon successful completion, the process group ID of the process with a process ID that matches pid shall be set to pgid.

As a special case, if pid is 0, the process ID of the calling process shall be used. Also, if pgid is 0, the process ID of the indicated process shall be used.

#### RETURN VALUE

Upon successful completion, setpgid() shall return 0; otherwise, -1 shall be returned and errno shall be set to indicate the error.

## ERRORS

The `setpgid()` function shall fail if:

**EACCES** The value of the `pid` argument matches the process ID of a child process of the calling process and the child process has successfully executed one of the `exec` functions.

**EINVAL** The value of the `pgid` argument is less than 0, or is not a value supported by the implementation.

**EPERM** The process indicated by the `pid` argument is a session leader.

**EPERM** The value of the `pid` argument matches the process ID of a child process of the calling process and the child process is not in the same session as the calling process.

**EPERM** The value of the `pgid` argument is valid but does not match the process ID of the process indicated by the `pid` argument and there is no process with a process group ID that matches the value of the `pgid` argument in the same session as the calling process.

**ESRCH** The value of the `pid` argument does not match the process ID of the calling process or of a child process of the calling process.

The following sections are informative.

## EXAMPLES

None.

## APPLICATION USAGE

None.

## RATIONALE

The `setpgid()` function shall group processes together for the purpose of signaling, placement in foreground or background, and other job control actions.

The `setpgid()` function is similar to the `setpgrp()` function of 4.2 BSD, except that 4.2 BSD allowed the specified new process group to assume any value. This presents certain security problems and is more flexible than necessary to support job control.

To provide tighter security, `setpgid()` only allows the calling process

to join a process group already in use inside its session or create a new process group whose process group ID was equal to its process ID. When a job control shell spawns a new job, the processes in the job must be placed into a new process group via `setpgid()`. There are two timing constraints involved in this action:

1. The new process must be placed in the new process group before the appropriate program is launched via one of the `exec` functions.
2. The new process must be placed in the new process group before the shell can correctly send signals to the new process group.

To address these constraints, the following actions are performed. The new processes call `setpgid()` to alter their own process groups after `fork()` but before `exec`. This satisfies the first constraint. Under 4.3 BSD, the second constraint is satisfied by the synchronization property of `vfork()`; that is, the shell is suspended until the child has completed the `exec`, thus ensuring that the child has completed the `setpgid()`. A new version of `fork()` with this same synchronization property was considered, but it was decided instead to merely allow the parent shell process to adjust the process group of its child processes via `setpgid()`. Both timing constraints are now satisfied by having both the parent shell and the child attempt to adjust the process group of the child process; it does not matter which succeeds first.

Since it would be confusing to an application to have its process group change after it began executing (that is, after `exec`), and because the child process would already have adjusted its process group before this, the `[EACCES]` error was added to disallow this.

One non-obvious use of `setpgid()` is to allow a job control shell to re-  
turn itself to its original process group (the one in effect when the job control shell was executed). A job control shell does this before returning control back to its parent when it is terminating or suspending itself as a way of restoring its job control "state" back to what its parent would expect. (Note that the original process group of the job control shell typically matches the process group of its parent, but this is not necessarily always the case.)

## FUTURE DIRECTIONS

None.

## SEE ALSO

`exec`, `getpgrp()`, `setsid()`, `tcsetpgrp()`

The Base Definitions volume of POSIX.1?2017, `<sys_types.h>`, `<unistd.h>`

## COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .

IEEE/The Open Group

2017

SETPGID(3P)