You can obtain the environment variables associated with the process by executing the following command:

```
$ cat /proc/12501/environ
GDM_KEYBOARD_LAYOUT=usGNOME_KEYRING_PID=1560USER=slynuxHOME=/home/slynux
```

Note that many environment variables are stripped off for convenience. The actual output may contain numerous variables.

The aforementioned command returns a list of environment variables and their values. Each variable is represented as a name=value pair and are separated by a null character (`\0`). If you can substitute the `\0` character with `\n`, you can reformat the output to show each variable=value pair in each line. Substitution can be made using the `tr` command as follows:

```
$ cat /proc/12501/environ  | tr '\0' '\n'
```

Now, let us see how to assign and manipulate variables and environment variables.

How to do it...

A variable can be assigned as follows:

```
var=value
```

`var` is the name of a variable and `value` is the value to be assigned. If `value` does not contain any space character (such as space), it need not be enclosed in quotes, Otherwise it is to be enclosed in single or double quotes.

Note that `var = value` and `var=value` are different. It is a usual mistake to write `var =value` instead of `var=value`. The later one is the assignment operation, whereas the earlier one is an equality operation.

Printing contents of a variable is done using by prefixing `$` with the variable name as follows:

```
var="value" #Assignment of value to variable var.


echo $var
```

Or:

```
echo ${var}
```

We will receive an output as follows:

```
value
```

We can use variable values inside `printf` or `echo` in double quotes:

```
#!/bin/bash
#Filename :variables.sh
fruit=apple
count=5
echo "We have $count ${fruit}(s)"
```

The output will be as follows:

```
We have 5 apple(s)
```

Environment variables are variables that are not defined in the current process, but are received from the parent processes. For example, `HTTP_PROXY` is an environment variable. This variable defines which proxy server should be used for an Internet connection.

Usually, it is set as:

```
HTTP_PROXY=192.168.1.23:3128
export HTTP_PROXY
```

The `export` command is used to set the `env` variable. Now any application, executed from the current shell script, will receive this variable. We can export custom variables for our own purposes in an application or shell script that is executed. There are many standard environment variables that are available for the shell by default.

For example, `PATH`. A typical `PATH` variable will contain:

```
$ echo $PATH
/home/slynux/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/usr/games
```

When given a command for execution, the shell automatically searches for the executable in the list of directories in the `PATH` environment variable (directory paths are delimited by the ":" character). Usually, `$PATH` is defined in `/etc/environment` or `/etc/profile` or `~/.bashrc`. When we need to add a new path to the `PATH` environment, we use:

```
export PATH="$PATH:/home/user/bin"
```

Or, alternately, we can use:

```
$ PATH="$PATH:/home/user/bin"
$ export PATH


$ echo $PATH
/home/slynux/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/usr/games:/home/user/bin
```