IEEE P1003.2a Draft 8 – December 1991

5.4 bg — Run jobs in the background

This utility shall be provided on systems that support both the User Portability 7
Utilities Option and the POSIX.1 {8} job control option. On other systems it is 7
optional. 7

5.4.1 Synopsis

`bg` [*job_id* . . .]

5.4.2 Description

If job control is enabled (see the description of `set –m` in 3.14.11), the `bg` utility shall resume suspended jobs from the current environment (see 3.12) by running them as background jobs. If the job specified by *job_id* is already a running background job, the `bg` utility shall have no effect and shall exit successfully.

Using `bg` to place a job into the background shall cause its process ID to become ''known in the current shell execution environment,'' as if it had been started as an asynchronous list; see 3.9.3.1.

5.4.3 Options

None.

5.4.4 Operands

The following operand shall be supported by the implementation:

job_id Specify the job to be resumed as a background job. If no *job_id* operand is given, the most recently suspended job shall be used. The format of *job_id* is described in 2.2.2.206.

5.4.5 External Influences

5.4.5.1 Standard Input

None.

5.4.5.2 Input Files

None.

5.4.5.3 Environment Variables

The following environment variables shall affect the execution of `bg`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments).

LC_MESSAGES This variable shall determine the language in which messages should be written.

5.4.5.4 Asynchronous Events

Default.

5.4.6 External Effects

5.4.6.1 Standard Output

The output of `bg` shall consist of a line in the format:

`"[%d] %s\n"`, <*job-number*> , <*command*>

where the fields are as follows:

<*job-number* > A number that can be used to identify the job to the `wait`, `fg`, and `kill` utilities. Using these utilities, the job can be identified by prefixing the job number with `%`.

<*command* > The associated command that was given to the shell.

5.4.6.2 Standard Error

Used only for diagnostic messages.

5.4.6.3 Output Files

None.

5.4.7 Extended Description

None.

5.4.8 Exit Status

The `bg` utility shall exit with one of the following values:

0 Successful completion.

>0 An error occurred.

5.4.9 Consequences of Errors

If job control is disabled, the `bg` utility shall exit with an error and no job shall be placed in the background.

5.4.10 Rationale. *(This subclause is not a part of P1003.2a)*

Usage, Examples

The `bg` utility will not work as expected when it is operating in its own utility execution environment because that environment will have no suspended jobs. In the following examples:

```
... | xargs bg
(bg)
```

each `bg` operates in a different environment and will not share its parent shell's understanding of jobs. For this reason, `bg` is generally implemented as a shell regular built-in.

History of Decisions Made

The extensions to the shell specified in POSIX.2 and POSIX.2a have mostly been based on features provided by the KornShell. The job control features provided by `bg`, `fg`, and `jobs` are also based on the KornShell. The developers of the standard examined the characteristics of the C-Shell versions of these utilities and found that differences exist. Despite widespread use of the C-Shell, the KornShell versions were selected for this standard to maintain a degree of uniformity with the rest of the KornShell features selected (such as the very popular command-line editing features).

8

The `bg` utility is expected to wrap its output if the output exceeds the number of display columns.

5.5 `crontab` — Schedule periodic background work

5.5.1 Synopsis

`crontab [`*file*`]`

`crontab [ −e | −l | −r ]`

5.5.2 Description

The `crontab` utility shall create, replace, or edit a user's *crontab* entry; a crontab entry is a list of commands and the times at which they are to be executed. The new crontab entry can be input by specifying *file* or input from standard input if no *file* operand is specified, or by using an editor, if `−e` is specified.

Upon execution of a command from a crontab entry, the implementation shall supply a default environment, defining at least the following environment variables:

HOME A pathname of the user's home directory.

LOGNAME The user's login name.

PATH A string representing a search path guaranteed to find all of the standard utilities.

SHELL A pathname of the command interpreter. When `crontab` is invoked as specified by this standard, the value shall be a pathname for `sh` (see 4.56).

The values of these variables when `crontab` is invoked as specified by this standard shall not affect the default values provided when the scheduled command is run.

If standard output and standard error are not redirected by commands executed from the crontab entry, any generated output or errors shall be mailed, via an implementation-defined method, to the user.

5.5.3 Options

The `crontab` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

`−e` Edit a copy of the invoking user's crontab entry, or create an empty entry to edit if the crontab entry does not exist. When editing is complete, the entry shall be installed as the user's crontab entry.