IEEE P1003.2a Draft 8 – December 1991

the digit string returned is a valid `day_number`, `year_number`, etc., based on the context.

5.3 batch — Execute commands when the system load permits

5.3.1 Synopsis

```
batch
```

5.3.2 Description

The `batch` utility shall read commands to be executed at a later time. It shall be the equivalent of the command:

```
at -q b -m now
```
7

where queue `b` is a special `at` queue, specifically for batch jobs. Batch jobs shall be submitted to the batch queue with no time constraints and shall be run by the system using algorithms, based on unspecified factors, that may vary with each 7
invocation of `batch`.

5.3.3 Options

None.

5.3.4 Operands

None.

5.3.5 External Influences

5.3.5.1 Standard Input

See 5.2.

5.3.5.2 Input Files

None.

5.3.5.3 Environment Variables

See 5.2.

5.3.5.4 Asynchronous Events

See 5.2.

5.3.6 External Effects

5.3.6.1 Standard Output

See 5.2. 8

5.3.6.2 Standard Error

See 5.2.

5.3.6.3 Output Files

None.

5.3.7 Extended Description

None.

5.3.8 Exit Status

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

- 0 The `batch` utility successfully submitted a job.
- >0 An error occurred.

5.3.9 Consequences of Errors

The job shall not be scheduled.

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

Usage, Examples

The `batch` utility reads from standard input the commands to be executed at a later time. It may be useful to redirect standard output within the specified commands.

This sequence can be used at a terminal:

```
batch
sort < file >outfile
EOT
```

This sequence, which demonstrates redirecting standard error to a pipe, is useful in a command procedure (the sequence of output redirection specifications is significant):

```
batch <<!
diff file1 file2 2>&1 >outfile | mailx mygroup
!
```

History of Decisions Made

Early drafts described `batch` in a manner totally separated from `at`, even though the historical model treated it almost as a synonym for `at -q b`. A number of features were added to list and control batch work separately from those in `at`. Upon further reflection, it was decided that the benefit of this was not worth the change in historical interface. It is anticipated that considerably more sophisticated ways of controlling background or batch work will be the subject of future revisions to this standard.

The −`m` option was included on the equivalent `at` command because it is historical 7
practice to mail results to the submitter, even if all job-produced output is 7
redirected. As explained in the rationale for `at`, the `now` keyword submits the job 7
for immediate execution (after scheduling delays), despite some historical systems 7
where `at now` would have been considered an error. 7

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.