

JOBS

JOBS (P)

trabajos manuales de POSIX programador (P)

NOMBRE

puestos de trabajo - estado de visualización de puestos de trabajo en el actual período de sesiones

SINOPSIS

empleos [-l | -p] [job_id ...]

DESCRIPCIÓN

La utilidad de los trabajos deberá mostrar el estado de los trabajos que se iniciaron en el shell actual

medio ambiente; ver Shell entorno de ejecución.

Cuando los trabajos informa el estado de terminación de un trabajo, el depósito deberá eliminar su ID de proceso

de la lista de los "conocida en el entorno de ejecución de shell actual ", véase asíncronos Listas Nous.

OPCIONES

La utilidad de los trabajos se ajustará al volumen Definiciones Base de IEEE Std 1003.1-2001, Sección 12.2, Directrices de sintaxis de utilidades.

Las siguientes opciones serán compatibles:

-l (La carta ell.) tiene más información acerca de cada trabajo de la lista. Esta información deberá incluir el número de trabajo, trabajo actual, ID de grupo de procesos, estado, y el comando que formaron el trabajo.

-p mostrar sólo los identificadores de proceso para los líderes de grupos de proceso de los trabajos seleccionados.

De forma predeterminada, la utilidad de los trabajos deberá mostrar el estado de todos los puestos de trabajo se detuvo, corriendo Back-

puestos de trabajo en tierra y todos los puestos de trabajo cuya situación ha cambiado y no se han reportado por el shell.

OPERANDOS

La siguiente operando Se apoyarán:

job_id Especifica los trabajos para los que el estado es que se mostrará. Si no se da job_id, se mostrará la información de estado para todos los trabajos. El formato de job_id es se describe en el volumen Definiciones Base de IEEE Std 1003.1-2001, Sección 3.203, Trabajo de Control de ID de trabajo.

STDIN

No se utiliza.

ENTRADA DE ARCHIVOS

Ninguno.

VARIABLES DE ENTORNO

Las siguientes variables de entorno afectarán a la ejecución de los trabajos:

LANG Proporcionar un valor predeterminado para las variables de internacionalización que son unset o

nulo. (Véase el volumen Definiciones Base de IEEE Std 1003.1-2001, Sección 8.2, Internacional

Variables de nacionalización de la precedencia de las variables de internacionalización utilizado

para determinar los valores de las categorías del local.)

LC_ALL Si se establece en un valor de cadena no vacía, anula los valores de todos los demás internacional

tionalization variables.

LC_CTYPE

Determinar la configuración regional para la interpretación de secuencias de bytes de datos de texto como

caracteres (por ejemplo, de un solo byte en lugar de multi-byte caracteres de argumentos).

LC_MESSAGES

Determinar la configuración regional que se debe utilizar para afectar el formato y contenido de diag-

mensajes de diag- escriben en el error estándar y los mensajes informativos escriben en estándares

salida dard.

NLSPATH

Determinar la ubicación de los catálogos de mensajes para el procesamiento de LC_MESSAGES.

Eventos asíncronos

Defecto.

STDOUT

Si el -p se especifica la opción, la salida consistirá en una línea para cada identificador de proceso:

"% D \ n", <ID de proceso>

De lo contrario, si el -l no se especifica la opción, la salida será una serie de líneas de la forma:

"[D%] % c % s % s \ n", <número-trabajo>, <actual>, <estado>, <comando>

donde los campos serán los siguientes:

<Actual>

El carácter '+' identifica el trabajo que se utilizaría como predeterminado para el fg o bg utilidades; este trabajo también se puede especificar mediante la job_id% + o "%%".

La

carácter '-' identifica el trabajo que se convertiría en el valor por defecto si la corriente trabajo por defecto fuera a salir; este trabajo también se puede especificar mediante la job_id% -. Para

otros puestos de trabajo, este campo es un <espacio>. A lo sumo un trabajo puede ser identificado con '+' y

a lo sumo un trabajo puede ser identificado con '-'. Si hay algún trabajo suspendido, a continuación,

el trabajo actual será un trabajo suspendido. Si hay al menos dos puestos de trabajo en suspensión,

entonces el trabajo anterior también será un trabajo suspendido.

<Trabajo-number>

Un número que puede ser utilizado para identificar el grupo de proceso a la espera, fg, bg, y

matar a los servicios públicos. El uso de estas utilidades, el trabajo puede ser identificado por el prefijo del

número de trabajo con '%'.

<Estado>

Una de las siguientes cadenas (en el entorno nacional POSIX):

Correr

Indica que el trabajo no ha sido suspendido por una señal y no ha salido.

Hecho

Indica que el trabajo completado y devuelto el código de salida cero.

Done (código)

Indica que el trabajo ha completado con normalidad y que salió con el especificado estado distinto de cero de salida, código, expresado como un número decimal.

Detenido

Indica que el trabajo fue suspendido por la señal SIGTSTP.

Detenido (SIGTSTP)

Indica que el trabajo fue suspendido por la señal SIGTSTP.

Detenido (SIGSTOP)

Indica que el trabajo fue suspendido por la señal SIGSTOP.

Detenido (SIGTTIN)

Indica que el trabajo fue suspendido por la señal SIGTTIN.

Detenido (SIGTTOU)

Indica que el trabajo fue suspendido por la señal SIGTTOU.

La aplicación puede sustituir la cadena suspendida en lugar de parado. Si el trabajo era terminado por una señal, el formato de <estado> no se especifica, pero deberá estar visiblemente dis-

tinct de todos los demás <estado> formatos de muestra aquí y deberá indicar el nombre o Descripción de la señal que causa la terminación.

<Comando>

El comando asociado que se le dio a la cáscara.

Si el -l se especifica la opción, se añadirá un campo que contiene el ID de grupo de procesos

antes de que el <estado> campo. También, más procesos en un grupo de procesos pueden ser de salida en separación

líneas de tarifas, utilizando sólo el ID de proceso y <comando> campos.

STDERR

Se utilizará el error estándar sólo para mensajes de diagnóstico.

SALIDA DE ARCHIVOS

Ninguno.

Descripción ampliada

Ninguno.

ESTADO DE SALIDA

Los siguientes valores de salida serán devueltos:

0 La conclusión con éxito.

> 0 Se ha producido un error.

CONSECUENCIAS DE ERRORES

Defecto.

Las siguientes secciones son informativos.

Uso de las aplicaciones

El -p opción es la única manera portátil para averiguar el grupo de proceso de un trabajo porque di-

implementaciones rentes tienen diferentes estrategias para definir el grupo de procesos de la trabajo. Uso como \$ (empleos -p) proporciona una forma de referirse al grupo de procesos del trabajo

de una manera independiente de la implementación.

La utilidad de los trabajos no funciona como se esperaba cuando se está operando en su propia ejecución utilidad

entorno de la causa que el medio ambiente no tiene puestos de trabajo aplicables a manipular. Vea el

Sección de uso de la aplicación para la bg. Por esta razón, los trabajos se implementa generalmente como una

bombardeo incorpora regularmente.

EJEMPLOS

Ninguno.

JUSTIFICACIÓN

Tanto "%%" y "% +" se utilizan para referirse al trabajo actual. Ambas formas tienen la misma validez

dad-la "%%" mirroring "\$\$" y "% +" que refleja la salida de puestos de trabajo. Ambas formas reflejan

la práctica histórica de la KornShell y el shell C con el control de trabajos.

Las funciones de control de trabajo proporcionadas por bg, fg, y puestos de trabajo se basan en el KornShell. La

desarrolladores estándar examinaron las características de las versiones del shell C de estas UTILIDADES

lazos y encontró que existen diferencias. A pesar del uso generalizado de la shell C, el Korn-

Versiones de Shell fueron seleccionados para este volumen de IEEE Std 1003.1-2001 para mantener un grado

de uniformidad con el resto de la KornShell características seleccionadas (tales como el muy popular

línea de comandos funciones de edición).

La utilidad de puestos de trabajo no depende de la opción de control de trabajo, al igual que el aparentemente relacionada

bg y utilidades fg porque los trabajos es útil para examinar trabajos en segundo plano, sin tener en cuenta

el estado de control de trabajos. Cuando el usuario ha invocado un conjunto + m comando y control de trabajo

se ha desactivado, trabajos todavía se pueden usar para examinar los trabajos de fondo asociados con

que la sesión actual. Del mismo modo, kill se puede usar entonces para matar a trabajos en segundo plano con kill%

<Fondo número de trabajo>.

La salida para los trabajos terminados se deja sin especificar para dar cabida a diferentes ma histórico

sistemas. Los siguientes formatos se han presenciado:

1. Killed (nombre de la señal)
2. Nombre de la señal
3. Nombre de la señal (coredump)

Descripción-núcleo 4. señal objeto de dumping

La mayoría de los usuarios deben ser capaces de entender estos formatos, aunque significa que las aplicaciones

tener problemas para analizar ellos.

El cálculo de las ID de trabajos no se describió ya que esto sugeriría una aplicación, que puede imponer restricciones innecesarias.

En una propuesta de principios, una -n opción se incluyó a "Visualizar el estado de los trabajos que tienen

cambiado, salido, o parado desde el último informe de situación ". Se fue eliminado por los cáscara siempre escribe cualquier cambio de estatus de empleo antes de cada pregunta.

DIRECCIONES FUTURAS

Ninguno.

VER TAMBIÉN

Shell Execution Environment, bg, fg, matar (), espera ()

DERECHOS DE AUTOR

Algunas partes de este texto se reimprimen y reproducidos en forma electrónica de IEEE Std 1003.1, Edición 2003, Norma para la Tecnología de la Información - Sistema operativo portable

Interface (POSIX), The Open Base Grupo de Especificaciones número 6, Copyright (C) 2001-2003 por

el Instituto de Ingenieros Eléctricos y Electrónicos, Inc y The Open Group. En el caso de discrepancia entre esta versión y la IEEE original y The Open Group

Estándar, el IEEE original y el estándar Open Group es el documento árbitro. El ginal

inal estándar se puede obtener en línea en <http://www.opengroup.org/unix/online.html>.

IEEE / The Open Group 2.003 puestos de trabajo (P)

NAME

jobs - display status of jobs in the current session

SYNOPSIS

jobs [-l] [-p][job_id...]

DESCRIPTION

The `jobs` utility shall display the status of jobs that were started in the current shell environment; see [Shell Execution Environment](#) .

When `jobs` reports the termination status of a job, the shell shall remove its `process ID` from the list of those "known in the current shell execution environment"; see [Asynchronous Lists](#) .

OPTIONS

The `jobs` utility shall conform to the [Base Definitions](#) volume of [IEEE Std 1003.1-2001](#),

Section 12.2, [Utility Syntax Guidelines](#).

The following options shall be supported:

-l (The letter ell.) Provide more information about each job listed. This information shall include the job number, current job, process group ID, state, and the command that formed the job.

-p Display only the process IDs for the process group leaders of the selected jobs.

By default, the `jobs` utility shall display the status of all stopped jobs, running background jobs and all jobs whose status has changed and have not been reported by the shell.

OPERANDS

The following operand shall be supported:

job_id Specifies the jobs for which the status is to be displayed. If no `job_id` is given, the status information for all jobs shall be displayed. The format of `job_id` is described in the [Base Definitions](#) volume of [IEEE Std 1003.1-2001](#), Section 3.203, [Job Control Job ID](#).

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of jobs:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of IEEE Std 1003.1-2001, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_CTYPE

Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

NLSPATH

Determine the location of message catalogs for the processing of LC_MESSAGES .

ASYNCHRONOUS EVENTS

Default.

STDOUT

If the **-p** option is specified, the output shall consist of one line for each process ID:

"%d\n", <process ID>

Otherwise, if the **-l** option is not specified, the output shall be a series of lines of the form:

"[%d] %c %s %s\n", <job-number>, <current>, <state>, <command>

where the fields shall be as follows:

<current>

The character '+' identifies the job that would be used as a default for the fg or bg utilities; this job can also be specified using the job_id %+ or "%%". The character '-' identifies the job that would become the default if the current default job were to exit; this job can also be specified using the job_id %-. For other jobs, this field is a <space>. At most one job can be identified with '+' and at most one job can be identified with '-'. If there is any suspended job, then the current job shall be a suspended job. If there are at least two suspended jobs, then the previous job also shall be a suspended job.

<job-number>

A number that can be used to identify the process group to the wait, fg, bg, and kill utilities. Using these utilities, the job can be identified by prefixing the job number with '%'. .

<state>

One of the following strings (in the POSIX locale):

Running

Indicates that the job has not been suspended by a signal and has not exited.

Done

Indicates that the job completed and returned exit status zero.

Done(code)

Indicates that the job completed normally and that it exited with the specified non-zero exit status, code, expressed as a decimal number.

Stopped

Indicates that the job was suspended by the SIGTSTP signal.

Stopped (SIGTSTP)

Indicates that the job was suspended by the SIGTSTP signal.

Stopped (SIGSTOP)

Indicates that the job was suspended by the SIGSTOP signal.

Stopped (SIGTTIN)

Indicates that the job was suspended by the SIGTTIN signal.

Stopped (SIGTTOU)

Indicates that the job was suspended by the SIGTTOU signal.

The implementation may substitute the string Suspended in place of Stopped. If the job was terminated by a signal, the format of <state> is unspecified, but it shall be visibly distinct from all of the other <state> formats shown here and shall indicate the name or description of the signal causing the termination.

<command>

The associated command that was given to the shell.

If the `-l` option is specified, a field containing the process group ID shall be inserted before the <state> field. Also, more processes in a process group may be output on separate lines, using only the process ID and <command> fields.

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

The `-p` option is the only portable way to find out the process group of a job because different implementations have different strategies for defining the process group of the job. Usage such as `$(jobs -p)` provides a way of referring to the process group of the job in an implementation-independent way.

The `jobs` utility does not work as expected when it is operating in its own utility execution environment because that environment has no applicable jobs to manipulate. See the APPLICATION USAGE section for `bg`. For this reason, `jobs` is generally implemented as a shell regular built-in.

EXAMPLES

None.

RATIONALE

Both `"%%"` and `"%+"` are used to refer to the current job. Both forms are of equal validity-the `"%%"` mirroring `"$$"` and `"%+"` mirroring the output of `jobs`. Both forms reflect historical practice of the KornShell and the C shell with job control.

The job control features provided by `bg`, `fg`, and `jobs` are based on the KornShell. The standard developers 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 volume of IEEE Std 1003.1-2001 to maintain a degree of uniformity with the rest of the KornShell features selected (such as the very popular command line editing features).

The `jobs` utility is not dependent on the job control option, as are the seemingly related `bg` and `fg` utilities because `jobs` is useful for examining background jobs, regardless of the condition of job control. When the user has invoked a `set +m` command and job control has been turned off, `jobs` can still be used to examine the background jobs associated with that current session. Similarly, `kill` can then be used to kill background jobs with `kill% <background job number>`.

The output for terminated jobs is left unspecified to accommodate various historical systems. The following formats have been witnessed:

1. Killed(signal name)
2. signal name
3. signal name(coredump)
4. signal description- core dumped

Most users should be able to understand these formats, although it means that applications have trouble parsing them.

The calculation of job IDs was not described since this would suggest an implementation, which may impose unnecessary restrictions.

In an early proposal, a -n option was included to "Display the status of jobs that have changed, exited, or stopped since the last status report". It was removed because the shell always writes any changed status of jobs before each prompt.

FUTURE DIRECTIONS

None.

SEE ALSO

Shell Execution Environment , bg , fg , kill() , wait()

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std

1003.1, 2003 Edition, Standard for Information Technology -- Portable Operating System

Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2003 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> .