

Comandos para GNU/Linux pocos conocidos

Hoy tuvimos el regocijo de leer [un artículo publicado por los colegas de LinuxAdictos](#) donde hablan sobre comandos pocos conocidos para la terminal GNU/Linux. Es por ello que decidimos publicar nuestra propia entrada pero dándole valor agregado: vamos a enfocarlo desde nuestra distribución preferida, Ubuntu Desktop (actualmente usamos las 16.04 LTS) y también como suplemento a [nuestra otra entrada sobre la introducción a la consola de comandos](#); ¡venid y acompañadnos en la búsqueda del conocimiento!

Comandos que vienen incluidos en Ubuntu.

Pues eso, comandos poco conocidos que ya vienen por defecto ya instalados en Ubuntu, nada más teclearlos los podemos utilizar. Lo que haremos acá es teclear el comando **"man"** seguido de un espacio y el nombre del comando a conocer, leeremos su descripción en inglés y las traduciremos y le buscaremos propósitos prácticos, comencemos pues.

Comando "calendar":

El comando **"calendar"** pertenece al paquete de **bsdmainutils** que viene por defecto en la distro GNU/Linux que utilizamos para esta entrada, Ubuntu 16.04 Xenial. Dado el caso extraño que no lo tengáis instalado podéis hacerlo con el comando **"sudo apt-get install bsdmainutils"** y sin ir a mucha profundidad en la materia, podréis ver los archivos que componen ese [paquete de software en este enlace](#).

Este paquete proviene del muy antiguo sistema Unix (el comando **"man calendar"** especifica que un comando **calendar** apareció en la ["Version 7 AT&T UNIX"](#) en 1979) y ciertamente que nos fue difícil encontrar a quien o quienes actualmente mantienen dicho repositorio, ya que el mero hecho de ser un calendario implica lo más dinámico de este mundo: el transcurrir del tiempo es siempre cambiante. En un "fork" al que tuvimos acceso hecho por Andrey Ulanov en el año 2002 comenta que proviene de **OpenBSD** e incluye una licencia que se hereda de cada uno de sus componentes, por ejemplo el módulo `pathnames.h` nos da la siguiente información:

```
/* $OpenBSD: pathnames.h,v 1.3 1996/12/05 06:04:41 millert Exp $ */
/* $
Id: pathnames.h,v 1.1.1.1 2002/11/26 23:37:38 drey Exp $ */
/* * Cop
yright (c) 1989, 1993 * The Regents of the University of California. Al
l rights reserved. * * Redistribution and use in source and binary fo
rms, with or without * modification, are permitted provided that the fo
llowing conditions * are met: * 1. Redistributions of source code mus
t retain the above copyright * notice, this list of conditions and the
following disclaimer. * 2. Redistributions in binary form must reproduc
```

```

e the above copyright      * notice, this list of conditions and the followi
ng disclaimer in the      * documentation and/or other materials provided wi
th the distribution.      * 3. All advertising materials mentioning features
or use of this software      * must display the following acknowledgement:
    * This product includes software developed by the University of      * Cal
ifornia, Berkeley and its contributors.      * 4. Neither the name of the Un
iversity nor the names of its contributors      * may be used to endorse or
promote products derived from this software      * without specific prior wr
itten permission.      *      * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CO
NTRIBUTORS ``AS IS'' AND      * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING
, BUT NOT LIMITED TO, THE      * IMPLIED WARRANTIES OF MERCHANTABILITY AND F
ITNESS FOR A PARTICULAR PURPOSE      * ARE DISCLAIMED. IN NO EVENT SHALL THE
REGENTS OR CONTRIBUTORS BE LIABLE      * FOR ANY DIRECT, INDIRECT, INCIDENT
AL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL      * DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS      * OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION)      * HOWEVER CAUSED AND ON AN
Y THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT      * LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY      * OUT OF THE US
E OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF      * SUCH DAMAGE
.      *      * @(#)pathnames.h 8.1 (Berkeley) 6/6/93      */

```

Como véis se remota hasta el año de 1989, época en la cual apenas nosotros entrábamos a estudiar en la universidad, ¡fascinante como pasa el tiempo! Otra cosa es la fuente de **bsdmainutils** en la distribución Debian donde el propio Ian Murdock introdujo código fuente según [se puede leer en esta página](#) (no obstante acá la reproducimos de igual manera):

```

This is a collection of programs from 4.4BSD-Lite that have not (yet) be
en re-written by FSF as GNU. It was constructed for inclusion in Debian
Linux. As programs found here become available from GNU sources, they
will be replaced. This package was originally put together by Austin D
onnelly
, but is heavily base
d on bsdutils
, originally put together by I
an Murdock
. Please report any problems or suggested additions or changes to Austi
n Donnelly. Then the package has been maintained by Charles Briscoe-Sm
ith . I gathered data for the 1999-2001 calendar files from various so
urces on the Internet, and I'd also like to thank Oliver Elphick, Julian
Gilbey, Daniel Martin and Jaldhar H. Vyas for providing much useful dat
a on the various religious calendars. I have edited the files they provi

```

ded to fit calendar's requirements, so any errors should be attributed to me. After cpbs@debian.org, Marco d'Itri maintained it for almost two years, before Tollef Fog Heen took over. The package is now maintained by Graham Wilson. This package may be redistributed under the terms of the UCB BSD license: Copyright (C) 1980 -1998 The Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Usando el comando "calendar".

Nada más en la terminal al introducir el comando sin ningún parámetro veremos algo similar esto, os indicamos que la salida por pantalla dependerá de la fecha de vuestro ordenador y os dará resultados similares o diferentes:

Figura 1:

Notad la última línea en castellano, ya os explicaremos luego. Como bien podéis observar la cosa va en idioma inglés lo cual no representa problema, tal cual el latín en el mundo antiguo fue la lengua franca así lo es hoy este idioma. *Lo que os advertimos es que hay varios idiomas más, tales como el francés y el ruso [justo arriba del Día de San Jorge esta en ruso el "[???????? ???? ?](#)" -aparición de Jesucristo a sus discípulos y la conversión del Apóstol Tomás-] y*

aunque no tiene calendarios del planeta entero, la intención del programa es bueno y tiene dos utilidades extras que estudiaremos pronto.

¿De dónde saca este programa **calendar** los datos a mostrar por pantalla? En la carpeta **/usr/share/calendar/** de nuestros ordenadores se almacenan una serie de ficheros con el nombre, cómo no, **calendar** pero con diferentes extensiones cada uno, por ejemplo **.russian**, **.christian** o **.computer**.

Figura 2

Como bien imaginamos por los metadatos, cada extensión describe lo que contiene el archivo y cuando llamamos al programa **calendar** simplemente lee y extrae los datos correspondientes a la fecha actual de cada fichero, pero hay más: también hay unos directorios con nombres claves que describen las dos primera letras el idioma separados de un guión bajo y dos letras más en mayúsculas que describen el país. Así tenemos que el directorio **de_AT** contiene en idioma alemán ("deuschland" en inglés) del país Austria (**AT**) donde se habla el alto alemán, más inclinado hacia el latín; de segundo en el listado está **de_DE** que corresponde al idioma Alemán de Alemania (ea, eso sonó feo en nuestro bello idioma castellano, nuestro idioma no es amistoso con esto de las redundancias).

Como somos unos "[hackers](#)" -o estamos en camino de 'graduarnos'- podemos hacer uso del comando **grep** y la 'tubería' "|" para filtrar resultados por una palabra clave específica, por ejemplo "Bonne fête" o "Feliz cumpleaños" en francés, para ello teclearemos lo siguiente: **calendar | grep "Bonne fête"** y solo tendremos dos líneas -o ninguna o varias, repetimos que eso depende de la fecha de nuestro ordenador-.

Esto es para búsquedas puntuales y exactas (claro que con el uso de expresiones regulares podremos ampliar la búsqueda -mayúsculas/minúsculas, caracteres comodín, etc), pero ¿qué tal si queremos ver temas específicos? Veamos alguno de estos ficheros de datos con fechas y busquemos uno que está en nuestro idioma, por ejemplo "**calendar.argentina**". Si le damos el comando **cat** en la ubicación **/usr/share/calendar** al fichero **calendar.argentina**:

Figura 3

Evidentemente escrito en lenguaje C notamos un encabezado describiendo el autor, unas variables de enlace para ser compiladas, el conjunto de caracteres a emplear y el conjunto de datos, uno por cada línea. Comienza con el mes abreviado en tres letras del idioma inglés, espacio, dos dígitos -cero a la izquierda de ser necesario-, **caracter tabulador** y luego la descripción del evento del correspondiente día.

Sumamente importante a destacar es que debemos separar la fecha de la descripción por medio del [caracter tabulador](#)(Tab?), de lo contrario la línea no será mostrada en el calendario (nuevo en esta versión y que se diferencia de versiones anteriores según "**man calendar**").

Hasta acá lo único especial son los comandos en lenguaje C ya que los eventos están muy bien "formateados" en texto plano pero son precisamente esas referencias en C las que nos indican que hay algo más. ***Esto quiere decir que "calendar" tiene una estructura rígida y que deben tener esos encabezados pero la cosa no termina allí.*** Si detalláis en la "Figura 2" nosotros agregamos un fichero **calendar.venezuela** con encabezado en toda regla y unos pocos acontecimientos históricos de nuestro país *pero nada de nada, no lo muestra para nada.* ***Eso quiere decir que solamente se atiende a los archivos que vengan con el paquete*** para probar esto último modificamos el fichero **calendar.unitedkingdom**:

```
sudo nano /usr/share/calendar/calendar.unitedkingdom
```

Y modificamos la línea del 23 de abril, Día de San Jorge, le colocamos la descripción en castellano y la muestra por pantalla sin problema alguno (figura 1). ***Lo bueno del software libre es que tenemos acceso directo al código fuente y podemos modificar y compilar, ya sea de manera estática o dinámica, a gusto con nuestras modificaciones.*** Pero antes de hacer esto sigamos estudiando las opciones ya establecidas en el comando "calendar" y luego veremos que más podemos hacer.

Parámetros del comando "calendar":

Primero podemos consultar los eventos de cualquier otra fecha simplemente acompañando de un espacio, el comando del parámetro "-t", otro espacio, y la fecha en formato AAAAMMDD, por ejemplo el 30 de diciembre de 2017 lo solicitamos de la siguiente manera:

```
calendar -t 20171230
```

Desglosamos la respuesta y no vemos por ningún lado la nota con el fallecimiento del gran **Ian Murdock**, padre de la distribución **Debian** (madre a su vez de Ubuntu) y que tiene su propio fichero **calendar.debian**. Por si acaso es que estamos cegatones volvemos a aplicar el comando **grep** de la siguiente manera:

```
calendar -t 20171230 | grep "urdock"
```

El cual no devuelve nada, aunque si usamos

```
calendar -t 20171230 | grep "KPD"
```

Sí que obtenemos respuesta: "Gründung der KPD, 1918", "Kommunistische Partei Deutschlands" o establecimiento del Partido Comunista Alemán, el 30 de diciembre de 1918. Es por ello que tecleamos el comando

```
sudo nano /usr/share/calendar/calendar.debian
```

y agregamos la siguiente línea (recordad de insertar pulsar la tecla Tabulador entre la fecha y la descripción):

```
Dec 30?Ian Murdock, stalwart proponent of Free Open Source Software, Father, Son, and the 'ian' in Debian.
```

Guardamos y listo, guardamos el recordatorio de tan lamentable fallecimiento para futuras generaciones ??.

Por defecto el comando **calendar** muestra dos días de acontecimientos (el nombre correcto son recordatorios) pero podemos especificar el número de días con el parámetro **-l** acompañado del número de días hacia adelante que queremos visualizar (nota: el otro parámetro que hace exactamente el mismo trabajo es **-A** pero como se presta a confusión con el otro parámetro **-a** usaremos siempre **-l**). Así si queremos ver exactamente los recordatorios del 30 de diciembre debemos teclear lo siguiente:

```
calendar -l 0 -t 20151230
```

Figura 4

Con el parámetro **-B** podemos especificar hacia atrás en el tiempo, desde la fecha señalada por el parámetro **-t**, el número de días que queramos ver. Por cierto que si se obvia el parámetro **-t** se toma la fecha del ordenador y si es viernes mostrará hasta el próximo lunes siguiente (si hay recordatorios); el parámetro **-w** se basa en eso para especificar el número de días hacia el futuro, pero no hayamos especialmente útil esa opción.

¿Recuerdan que comentamos haber realizado un calendario para Venezuela? Pues bueno llegó la hora de poder visualizarlo con el parámetro **-f** acompañado de la ruta de ubicación del archivo de marras, para ello metemos:

```
calendar -t 20170101 -l 364 -f /usr/share/calendar/calendar.venezuela
```

Figura 5

¿Qué tal nos ha quedado? Para subir este archivo y contribuir con el software libre publicaremos dicho archivo en GitHub con licencia "Creative Commons" e instrucciones sobre como instalarlo y utilizarlo.

Instalando "calendar.venezuela" en nuestro ordenador.

Según las instrucciones en el archivo **calendar.all** *deberíamos* copiar dicho archivo a nuestra carpeta personal "**~/calendar/**" si queremos editar dicho archivo. **Esto es así ya que si compartimos un ordenador con varias personas lo lógico es que cada quien tenga su calendario particular, es decir, en el fichero "calendar.all" podemos especificar cuales ficheros queremos ver y/o utilizar.** Si lo copiamos a nuestro perfil personal podemos editarlo *sin molestar a los otros usuarios del sistema.*

En nuestro caso decidimos modificar el archivo **calendar.all** para todos los usuarios para que tengan acceso al fichero de recordatorio de Venezuela (ambos en la carpeta **/usr/share/calendar/**). Para ello agregamos al archivo la siguiente línea, justo al final, por orden alfabético:

```
#include
```

Luego guardamos y podemos inmediatamente comprobar si funciona. De igual manera así como agregamos, podemos "quitar" ficheros de recordatorios que no pensamos usar. En realidad quitamos es la referencia o referencias, no el o los archivos en sí mismos.

Comandos "cal" y "ncal":

Si necesitamos ver por pantalla el mes actual simplemente tecleamos **cal** y lo veremos por pantalla y el día actual estará resaltado. De manera similar funciona **ncal** pero en vez de tener los días de la semana en el eje de las abscisas los coloca en el eje de las ordenadas. Los parámetros son comunes a ambos comandos, pero tienen sus excepciones (os queda como tarea averiguar cuáles):

- Si no queremos ver resaltado el día actual usaremos el parámetro **-h**.
- Si queremos ver un mes específico del año actual usaremos **-m n** donde **n** es el número del mes. También podemos colocar las tres primeras letras del mes (inglés o castellano, sin embargo no reconoce francés ni ruso).
- Si queremos ver el año completo -en curso- colocamos **-y**. Si necesitamos un año específico lo colocamos al lado (calendario gregoriano). Por ejemplo podremos saber que el 26 de diciembre del año 7017 será un día viernes. ?
- Podremos combinar mes y año en la misma orden, ejemplo: **cal -m 12 2117** mostrará el mes de diciembre del año 2117.
- Algo muy útil es visualizar el mes anterior, el actual y el venidero: para ellos usamos el parámetro **-3**.
- Para ver varios meses hacia adelante, **-A** seguido del número de meses; **-B** especifica los meses hacia atrás. Ejemplo: "**cal -B 1 -A 3**".
- Como sabemos el día cero en informática es el domingo, día del Señor, quien al séptimo día tomó descanso (sábado para los judíos). Para nosotros los católicos el domingo es día de adorar a Dios así que comenzamos a trabajar los lunes, por ello podemos especificar que las semanas comiencen los días lunes con **-M** y con **-S** comienzan los domingos (opción por defecto). Esta opción no funciona con **cal** no obstante **ncal** tiene un parámetro **-b** que tiene el mismo estilo de **cal**, ¡probad!

Comando "yes":

Brevemente diremos que es un comando para nosotros los "nerds": el comando **yes** repite indefinidamente por pantalla el texto que lo acompañe. Para detenerlo debemos usar la combinación de teclas **CTRL+C** (esto es válido para todos los comandos de terminal). Viene con el [conjunto de utilerías básicas GNU](#).

Comando "col":

El comando **col** también es poco conocido pero tiene el aliciente de trabajar con los comandos **nroff**, **tbl** y **colcrt**; sin embargo le hallamos cierta utilidad didáctica si lo combinamos con los caracteres especiales utilizados para el **shell** y el **lenguaje C**. Es por ello que comenzaremos de atrás para adelante: presentamos los caracteres especiales y terminamos con el comando **col**.

Pues bien, en el juego de caracteres ASCII (ahora sustituidos por los caracteres UNICODE y representados de manera práctica con UTF-8 [-revisad nuestro tutorial sobre HTML5-](#)) los primeros 31 caracteres son especiales, llamados "de sistema" y tienen diversos usos especiales. Como abre boca tenemos el **ACK** (acknowledgement) o reconocimiento: si recibimos un **datagrama** por **TCP** (o un paquete por puerto serial) y el contenido es recibido de conformidad a reglas preexistentes *simplemente contestamos enviando dicho caracter, el número 6 en sistema de numeración decimal (en sistema binario es **000 0110** e igual es 6 en octal y hexadecimal)*. Además de esta numeración en el lenguaje C (así como Java, Perl e incluso Visual Basic) se le representa con **caracteres de escape** (esto se presta a confusión a la hora de traducir al castellano, ojito como lo interpretáis). Estos **caracteres de escape** tienen la función de salir, *escaparse* de la representación tal como son y se interpretan por pantalla e impresoras de matriz de punto (y algunas de inyección de tinta) de manera diferente. *¿Qué diantres tiene esto que ver con la terminal GNU/Linux o "shell"? ?*

Vamos a la práctica directamente con solamente un caracter: el número 11 que corresponde al **tabulador vertical** (no confundir con el **tabulador horizontal**) al cual podemos almacenar en una variable con ayuda del operador **\$** el cual interpreta los códigos "escapados" por medio de **"\0"** -si es octal- o **"\x"** si es hexadecimal. En este caso sería **"\013"** en octal o **"\x0b"** en hexadecimal. Corred los siguientes ejemplos en vuestra terminal:

```
tab_ver=$'\013'  echo "$tab_ver"  tab_ver=$'\x0B'  echo "$tab_ver"
```

Como notaréis *aparentemente* no vemos nada por pantalla y para notar como trabaja el tabulador vertical podemos acompañarlo de texto descriptivo de la siguiente manera, mirad la animación:

Allí veís el famoso comando **col** en acción. Otra forma de representar al tabulador vertical en lenguaje C es con **"\v"**:

```
jimmy@KEVIN:~$ linea=$'\vLinea 1.\vLinea 2.\vLinea 3.'  jimmy@KEVIN:~$ ec  
ho "$linea"      Linea 1.          Linea 2.          Linea 3.  jimmy@KEVIN:~$
```

Comando "colrm":

El comando **colrm** nos permite *remover* la primera *n columnas* ya sea de un archivo de texto o lo que ingresemos por consola de comandos. Hacemos notar que la numeración comienza en uno y no en cero como la mayoría de las matrices en computación. El ejemplo aclara dudas:

Otro uso es suministrarle dos parámetros: la columna de inicio y el largo que removerá:

Comando "column":

El comando **column** viene a ser de utilidad si queremos mostrar en columnas una lista que tengamos ya sea en un archivo o en una entrada de texto de terminal con el conector tubería "|".

Comando "from":

Este comando permite visualizar lo que hallan enviado los usuarios por el sistema de correo en un ordenador GNU/Linux. Sirve como sistema de comunicación entre los usuarios registrados en un ordenador.

Comando "hexdump":

Permite desglosar los códigos ASCII en su representación decimal, octal o hexadecimal e incluso aplicándole un formato específico (aparte de los que trae de manera predeterminada) ya sea de un archivo o por entrada de terminal. Aquí vemos el formato más útil para dicho comando, pero las posibilidades son infinitas:

```
jimmy@KEVIN:~$ echo "ABCD" | hexdump -v -e '/1 "%_ad# "' -e '/1 "%02X hex  
"' -e '/1 " = %03i dec"' -e '/1 " = %03o oct"' -e '/1 " = _%c\_\\n"' 0# 4  
1 hex = 065 dec = 101 oct = _A_ 1# 42 hex = 066 dec = 102 oct = _B_ 2#  
43 hex = 067 dec = 103 oct = _C_ 3# 44 hex = 068 dec = 104 oct = _D_ 4#  
0A hex = 010 dec = 012 oct = _
```

Esencialmente usamos el comando **echo** para enviar las cuatro primeras letras del abecedario por medio del comando "tubería" "|" y entregárselo a **hexdump** el cual lo desglosa carácter a carácter y muestra su código ASCII en diferentes sistemas de numeración para luego insertar un retorno de carro al final de cada línea "\n".

Comando "look":

Si por ejemplo necesitamos buscar palabras que tengan un prefijo común, por ejemplo "oft", introducimos **"look oft"** lo cual devuelve unas cuantas palabras... en inglés (**often**: a menudo, frecuentemente). Si bien indica que depende de la configuración que tengamos en **ENVIRON** (por ejemplo podemos ver cual lenguaje tenemos configurado si tecleamos **echo \$LANG**) siempre utiliza el idioma inglés por defecto. Para buscar una lista de palabras en idioma castellano

deberemos especificar que usaremos el diccionario ubicado en **/usr/share/dict/spanish**. ¿Utilidad? Tal vez querramos hacer un ataque de fuerza bruta a una contraseña usando palabras comunes, quien sabe.

Comandos "wall" y "write":

Con **wall** podremos enviar un mensaje a todos los que estén conectados o conectadas por consola, quieran o no recibir el mensaje. Por el contrario, si queremos enviar a un usuario específico usamos el comando **write** acompañado del nombre de usuario y la terminal donde esté conectado. Para saber quiénes están en línea usaremos **who**.

<https://www.youtube.com/watch?v=4yX29R81doY>

Otros comandos poco conocidos:

lorder, ul.

Comandos que se deben instalar desde repositorios oficiales.

Comando "termsave":

Comando "pv":

Comando "banner":

Fuentes consultadas:

En idioma castellano:

- "[Comandos para Linux algo desconocidos que deberías conocer](#)" por Isaac P.E.
- "[Caracter Tabulador](#)" en Wikipedia.
- "[Repositorio para «calendar.venezuela»](#)" por Jimmy Olano en GitHub.

En idioma inglés:

- "[Version 7 AT&T Unix](#)" at Wikipedia.
- "[Calendar program not showing Father's Day](#)".
- "[Debian mourns the passing of Ian Murdock](#)" at Debian.org
- "[How To View Significant Events In World History From Command line](#)".
- "[Linux and Unix calendar command](#)".
- "[File list of package bsdmainutils in xenial of architecture amd64](#)" at Ubuntu.com
- "[Coreutils - GNU core utilities](#)".

- "[Special Characters](#)" Advanced Bash-Scripting Guide.
- "[colrm Command](#)" IBM Knowledge Center.
- "[UNIX / Linux look Command Examples](#)" by SathiyaMoorthy.
- "[Linux: List All Environment Variables Command](#)" at nixCraft.
- "[Write a message to login users through terminal](#)".

En idioma ruso:

- "[????? ????-GUI ??? ?????????? ?? ???????????](#)" ?? Nugensovet.