



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Resultados Linux Trivial Pursuit: 1 - 10 (20649 lectures)

Per Ricardo Galli Granada, [gallir](http://mnm.uib.es/gallir/) (<http://mnm.uib.es/gallir/>)

Creado el 23/03/2001 15:42 modificado el 23/03/2001 15:42

*Para matar el aburrimiento, estos días hemos estado jugando al **Linux Trivial Pursuit** en la lista de correo Bulmailing. Simplemente hacíamos algunas preguntas complicadillas o poco usuales para ver quién contestaba primero. Aunque el objetivo principal era que todos aprendamos un poco y conocer alternativas. Estas son las 10 primeras preguntas, ganadores y respuestas correctas (aunque en algunos casos, no las únicas).*

Pregunta 1

¿Como se puede saber, sin acceder e ningún fichero en /etc y sin saber si se está en el modo default o no, en que "runlevel" está el sistema?

Respuesta

Hay varias respuestas correctas: [Mateu](#)⁽¹⁾, [Javi](#)⁽²⁾, [MAC](#)⁽³⁾. Ejecutando el comando `runlevel` o mirando si el proceso `init` tiene un argumento numérico. El número indica el runlevel.

Pregunta 2

¿Como se hace para colgar, bestialmente, el Linux con un (1) comando?

Respuesta

Javi fué el primero en [responder](#)⁽⁴⁾ con una bastante cercana. Hay varias, la "menos" dañina es

```
cp /dev/zero /dev/mem
```

Pregunta 3

*¿Como se hace para que dos o mas procesos lean `_cualquier_` caracter de un fichero sin usar ninguna de las funciones `*seek` y que además el valor leído sea el último escrito por otro proceso?*

- Se permite un solo `open`.
- Si se modifica el fichero con `cat`, `vi`, etc., el cambio debe ser reflejado en los procesos que leen.
- Se parece a memoria compartida...

Respuesta

Usando el `mmap`. Ganador [Mateu](#)⁽⁵⁾.



Pregunta 4 (La [hizo y la contestó](#)⁽⁶⁾ Daniel R. C.)

¿Para que sirven la mayoría de los comandos de las aplicaciones Linux?

Respuesta

Para impresionar a alguien que nos esta mirando por encima del hombro XDD Extraido de "Guia de Referencia y Aprendizaje de Linux" Ed: O'Reilly

Pregunta 5

Si tenéis miles y miles de fichero en un directorio (típico de animación por ordenador o edición de vídeo) y tenéis que copiar o mover (o ejecutar cualquier programa) enviando todos los nombres como argumentos. Algo del tipo:

```
mv *.jpg /xxx
```

¿Como lo hacéis?

Respuesta

Hay varias muy parecidas. Una la presentó [Guillem](#)⁽⁷⁾, usando el `xargs` y otra [Mateu](#)⁽⁸⁾ usando el `find`. Esta es mi respuesta

```
ls | grep /\.jpg | xargs comando opcion
```

o para agregar un argumento al final en el caso de un `cp` o `mv`:

```
ls | awk '/\.jpg$/{print $_}; END {print "DIR_DESTINO"}' | xargs mv
```

Pregunta 6

¿Como se hace para poder ver la salidas estándar de un programa por la pantalla y además grabarlo en un/os ficheros?

Respuesta

El ganador fue [Mateu](#)⁽⁹⁾. Por ejemplo:

```
ls | tee unacopia otracopia | less
```

Pregunta 7

¿Cual es el tamaño del buffer interno del Linux para un pipe entre dos procesos? (son iguales para el 2.2 y el 2.4)



Respuesta

El ganador fue [Miguel Ángel Coll](#)⁽¹⁰⁾, son 4096 bytes. Luego [Javi Polo](#)⁽¹¹⁾ nos mostró donde está definida:

```
en un 2.4 he encontrado esto en limits.h:
#define PIPE_BUF      4096      /* # bytes in atomic write to a pipe */
```

Pregunta 8

Esta es una compuesta:

1. ¿Cuántas llamadas de sistemas tiene el 2.4?
2. ¿Cuántas más tiene con respecto al 2.2?
3. ¿En qué fichero están definidas?

Respuesta

El ganador ha sido [Miguel Ángel Coll](#)⁽¹²⁾. El 2.4 tiene 221 llamadas de sistema, el 2.2 tiene 190 y están definidas en el fichero `include/asm-i386/unistd.h`.

Pregunta 9

¿Que han hecho estos señores que los hace conocidos en la comunidad Linux?

R. Card, W. Davidson.

¿Quiénes les han ayudado a mejorarlo?

Respuesta

El ganador es [Javi Polo](#)⁽¹³⁾. Rémy Card diseñó el `ext`, que era mucho más lento que el `MinixFS` y producía mucha fragmentación. Entonces, con la ayuda de Wayne Davidson y además Stephen Tweedie y Theodore Ts'o diseñaron el `ext2` que se convirtió en el sistema estándar del Linux.

Pregunta 10

Este código es parte de la inicialización del kernel. ¿Que hace?

```
pushw      %cx
movb       $0x02, %ah
int        $0x1a
movb       %dh, %al
andb       $0x0f, %al
movb       %dh, %ah
movb       $0x04, %cl
shrb      %cl, %ah
aad
popw      %cx
ret
```



Respuesta

El ganador ha sido [Tomeu](#)⁽¹⁴⁾. Javi también ha [contestado](#)⁽¹⁵⁾ correctamente. Lee la hora del reloj CMOS cuando el Linux arranca.

Lista de enlaces de este artículo:

1. <http://bulma.net/pipermail/bulmailing/2001-March/000726.html>
 2. <http://bulma.net/pipermail/bulmailing/2001-March/000706.html>
 3. <http://bulma.net/pipermail/bulmailing/2001-March/000701.html>
 4. <http://bulma.net/pipermail/bulmailing/2001-March/000770.html>
 5. <http://bulma.net/pipermail/bulmailing/2001-March/000768.html>
 6. <http://bulma.net/pipermail/bulmailing/2001-March/000793.html>
 7. <http://bulma.net/pipermail/bulmailing/2001-March/000802.html>
 8. <http://bulma.net/pipermail/bulmailing/2001-March/000804.html>
 9. <http://bulma.net/pipermail/bulmailing/2001-March/000816.html>
 10. <http://bulma.net/pipermail/bulmailing/2001-March/000878.html>
 11. <http://bulma.net/pipermail/bulmailing/2001-March/000890.html>
 12. <http://bulma.net/pipermail/bulmailing/2001-March/000883.html>
 13. <http://bulma.net/pipermail/bulmailing/2001-March/000893.html>
 14. <http://bulma.net/pipermail/bulmailing/2001-March/000884.html>
 15. <http://bulma.net/pipermail/bulmailing/2001-March/000894.html>
-

E-mail del autor: gallir_ARROBA_uib.es

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=586>