

HISTORIAL DE REVISIONES			
NÚMERO	FECHA	MODIFICACIONES	NOMBRE

Tema 4 - Sistemas de paquetes (DCA)

Índice

1. Distribución eficiente del software.	1
2. Algunos formatos de paquetes para distribuir software	1
3. Formatos TAR / TGZ / TBZ / TXZ (I)	1
4. Formatos TAR / TGZ / TBZ / TXZ (II)	2
5. Formatos TAR / TGZ / TBZ / TXZ (III)	2
6. Formatos TAR / TGZ / TBZ / TXZ (IV)	2
7. Formatos TAR / TGZ / TBZ / TXZ (V)	3
8. Formato DEB (I)	3
9. Formato DEB (II)	3
10. Formato DEB (III)	4
11. Formato DEB (IV)	4
12. Formato DEB (V)	4
13. Formato DEB (VI)	5
14. Formato DEB (VII)	5
15. Formato DEB (VIII)	6
16. Formato RPM	6
17. Aplicación alien	6
18. Trabajo en grupo en clase	7
19. Práctica individual	7
20. Aclaraciones	7

Logo DLSI

Tema 4 - Sistemas de paquetes Curso 2018-2019

1. Distribución eficiente del software.

- Una vez tenemos nuestro software listo para ser utilizado...
- Debemos hacerlo llegar a los potenciales usuarios...
- Cuanto más fácil de instalar les sea... más probable es que lo empleen.
- Debemos evitar instalaciones tipo **Windows**... donde el s.o. no nos proporciona los mecanismos de gestión apropiados para la instalación de software...
 - Desconoce componentes ya instalados.
 - No tiene en cuenta dependencias
 - Permite la sobrescritura de archivos ya instalados (**DLL Hell**)
 - La propia Microsoft ofrece como alternativa a estos instaladores el conocido como **Windows Installer**, para la instalación, mantenimiento y eliminación de programas en s.o. de la familia Windows.

2. Algunos formatos de paquetes para distribuir software

VAMOS A VER A PARTIR DE ESTE PUNTO ALGUNOS DE LOS FORMATOS DE DISTRIBUCIÓN DE SOFTWARE EN EL MUNDO DEL *software libre*.

- TAR / TGZ / TBZ / TXZ
- DEB
- RPM

3. Formatos TAR / TGZ / TBZ / TXZ (I)

- El formato original es TAR.
- Es un formato de archivos ampliamente usado en entornos UNIX.
- Los archivos en este formato usan la extensión `.tar`
- Es un formato pensado inicialmente para ser usado con cintas magnéticas, de ahí su nombre Tape **AR**chiver (en inglés: archivador en cinta)
- `Tar` sólo reúne los diferentes ficheros en un solo archivo, no realiza compresión de datos, por tanto no reduce el tamaño de los archivos que crea.

4. Formatos TAR / TGZ / TBZ / TXZ (II)

- La órden de llamada suele tener este formato:

```
tar <<opciones>> <<archivoSalida>> <<archivo1>> ... <<archivoN>>
```

- Dado que "tar" fue diseñado en sistemas "UNIX", almacena la metainformación de los ficheros en estos s.o.: fecha de última modificación, propietario y permisos al estilo UNIX. Además admite el almacenamiento de *enlaces simbólicos* y *enlaces duros*
- El formato tar está estandarizado por las normas **POSIX1** y posteriormente por **POSIX2**.
- Veamos algunos ejemplos prácticos de uso...

5. Formatos TAR / TGZ / TBZ / TXZ (III)

- Hemos dicho que tar no comprime...solo empaqueta.
- Pero es bastante habitual comprimir después de empaquetar...
- ...por eso permite enlazar con diversos compresores de forma automática:

- gzip... usando la opción **-z**
- bzip2... usando la opción **-j**
- xz... usando la opción **-J**

- Por tanto,un archivo tar comprimido con:

- gzip puede usar las extensiones: ".tar.gz" o ".tgz"
- bzip2 puede usar las extensiones: ".tar.bz" o ".tbz"
- xz puede usar las extensiones: ".tar.xz" o ".txz"

6. Formatos TAR / TGZ / TBZ / TXZ (IV)

- Hagamos una serie de pruebas en modo texto y modo gráfico...
- A pesar de lo sencillo de usar que hemos visto que es el formato tar...
- Además de ser un estandar POSIX...
- ...resulta que tiene una serie de restricciones/carencias para ser usado en la distribución de software de manera eficiente...
- Al distribuir software es bastante normal que no baste solo con copiar determinados archivos en lugares concretos..., por ejemplo:
 - ...que tengamos que ejecutar determinadas acciones para finalizar la instalación correctamente.
 - ...que nos interese saber si hay instalado un determinado software previamente y en alguna versión concreta (comprobación de dependencias)...

7. Formatos TAR / TGZ / TBZ / TXZ (V)

- Es por eso que en la distribución de software, por parte de determinados s.o., se suelen emplear otros formatos tales como **DEB** y **RPM**.
- En estos ficheros, además de la información a instalar, podemos indicar:
 - acciones a llevar a cabo como parte del proceso de instalación/desinstalación.
 - comprobación de dependencias de software previamente instalado
 - software recomendado que complementa al que vamos a instalar y puede ser de ayuda al usuario.
- En el resto del tema nos centraremos en el formato DEB y veremos como generar RPM desde un DEB de manera semi-automática.

8. Formato DEB (I)

- La estructura interna de estos ficheros es la de un archivo **ar**.
- Internamente contiene tres archivos:
 - "debian-binary": fichero de texto que contiene el número de versión del formato deb. Actualmente es la 2.0.
 - "control.tar.gz": Contiene la *meta-información* del paquete.
 - "data.tar", "data.tar.gz", "data.tar.bz2" o "data.tar.lzma": Contiene los archivos a instalar.

9. Formato DEB (II)

Veamos como crear un archivo en formato DEB de forma manual. Cuando terminemos verás que este archivo se parece más a un TAR que a un DEB ya que la parte de *control* no estará completa.

- Estando en el directorio principal de nuestro proyecto, ejecutamos:

```
mkdir ./debian
```

- Bajo el subdirectorio ./debian reproducimos toda la estructura de directorios de nuestro software cuando esté instalado. En este ejemplo sencillo solo tenemos los subdirectorios "usr" y "usr/bin".
- Estando en el directorio principal de nuestro proyecto, ejecutamos:

```
mkdir -p ./debian/usr/bin
```

- Copiamos manualmente los ficheros que componen nuestro software a los directorios que corresponda bajo "./debian":

```
cp helloworld ./debian/usr/bin
```

10. Formato DEB (III)

- Creamos el directorio `./debian/DEBIAN` y dentro de él, el fichero `./debian/DEBIAN/control`.
- Este fichero tiene este formato:

```
1 Package: helloworld
2 Priority: optional
3 Section: misc
4 Maintainer: dca
5 Architecture: amd64
6 Version: 1.0
7 Depends:
8 Description: Increible implementacion de HolaMundo
9 Al ser ejecutado, imprime una linea en el terminal con el
10 texto: 'Hola Mundo'.
```

- Y ahora solo nos queda construir el fichero DEB:

```
fakeroot dpkg -b ./debian /ruta/hasta/fichero.deb
```

11. Formato DEB (IV)

- El ejemplo anterior ha sido a nivel ilustrativo...
- En la práctica dispondremos, por ejemplo, del código fuente del software en un fichero TGZ y tendremos que compilarlo y seguir los pasos que marque su creador para tener una versión instalable del mismo.
- Afortunadamente en estos casos, todo ello se gestiona con uno o varios ficheros Makefile, los cuales disponen de un objetivo o target llamado `install` que copia los archivos necesarios a los directorios correspondientes.
- Afortunadamente también, los creadores del formato DEB nos proporcionan algunas utilidades para facilitar esta labor.
- Es el caso de la aplicación `dh_make`, según indica en su página de manual:

```
dh_make - prepare Debian packaging for an original source archive
```

- Veamos con un ejemplo el funcionamiento de `dh_make`

12. Formato DEB (V)

Tomamos como ejemplo la aplicación `helloworlddca` descargada de `helloworlddca-1.0.tgz`.

Su contenido es un directorio: `helloworlddca-1.0` , que contiene los archivos: `hw.c` y `Makefile`

```
1 #include <stdio.h>
2 int main (int argc, char *argv[]) {
3     printf ("Hola Mundo\n");
4     return 0;
5 }
```

13. Formato DEB (VI)

```
1 # Makefile
2
3 DESTDIR=.
4 PROG=hw
5
6 all: $(PROG)
7
8 $(PROG): $(PROG).c
9     gcc $(PROG).c -o $@
10
11 install: $(PROG)
12     mkdir -p $(DESTDIR)/usr/bin
13     cp $(PROG) $(DESTDIR)/usr/bin/
14
15 clean:
16     rm -f *~ $(PROG) *.o
17
18 .PHONY: clean install all
```

14. Formato DEB (VII)

- Dentro del directorio `helloworlddca-1.0` ejecutamos:

```
dh_make -s -e mail@provider.com -f ../helloworlddca-1.0.tgz
```

Esto genera el directorio `"debian"` con este contenido:

changelog	docs	init.d.ex	menu.ex	prerm.ex
compat	helloworlddca.cron.d.ex	manpage.1.ex	postinst.ex	README.Debian
control	helloworlddca.default.ex	manpage.sgml.ex	postrm.ex	README.source
copyright	helloworlddca.doc-base.EX	manpage.xml.ex	preinst.ex	rules
source	watch.ex			

- La extension `.ex` indica que se trata de ejemplos, si los usamos, la quitamos y modificamos el contenido de estos archivos según corresponda.
- Destacamos los archivos `"changelog"`, `"control"`, `"rules"` y los `"pre|post inst"` y `"pre|post rm"`.
- Modifiquemos su contenido y pasemos a generar el archivo `" .deb "`. Para ello usamos la orden:

```
dpkg-buildpackage -rfakeroot -us -uc
```

Las opciones **-us** (unsigned source, no firma el paquete fuente con gpg) y **-uc** (unsigned changes, no firma los cambios con gpg) son necesarias si no tenemos configurado gpg.

15. Formato DEB (VIII)

- Conforme vayamos modificando los archivos que controlan el empaquetado en formato `.deb`, debemos:
 - Crear una nueva entrada en `debian/changelog`, opcionalmente incrementado la versión del empaquetado.
 - Regenerar el archivo `.deb`.
- Para crear una nueva entrada en `debian/changelog` disponemos de la aplicación `"dch"`.
- Es **importante** tener en cuenta que `"dh_make"` hace uso de los programas que se encuentran en el paquete `"debhelper"`.
- Una vez tenemos generado un `".deb"` lo podemos instalar --como administradores-- mediante la orden **`dpkg`**:

```
dpkg -i helloworlddca_1.0-1_amd64.deb
```

- O pedir información sobre el mismo con:

```
dpkg -I helloworlddca_1.0-1_amd64.deb
```

Importante

"dpkg" no tiene en cuenta dependencias con otros paquetes...para ello necesitas otras aplicaciones como `"apt-get"` o `"aptitude"` o `"apt"`.

16. Formato RPM

- Se trata de un formato similar al `.deb`.
- Se emplea en distribuciones como Red Hat Linux, Fedora Linux, SuSE Linux, etc...
- No vamos a ver directamente como crear paquetes en este formato...pero sí que lo vamos a hacer de forma semi-automática con la aplicación **`alien`**.

17. Aplicación alien

- Se ejecuta como administrador o con la orden `sudo`.
- Se le pasa como parametro el nombre del archivo `.tgz`, `.deb` o `.rpm` a convertir de formato...
- ...y una opción que le indica el formato destino:
 - `-t` convierte de `".deb"` o `".rpm"` a `".tgz"`
 - `-r` convierte de `".deb"` o `".tgz"` a `".rpm"`
 - `-d` convierte de `".tgz"` o `".rpm"` a `".deb"`
- En nuestro caso, una vez hemos generado el `.deb` podemos hacer:

```
sudo alien -t ../helloworlddca_1.0-1_amd64.deb
sudo alien -r ../helloworlddca_1.0-1_amd64.deb
```

- `alien` dispone también de la opción `-c`, la cual trata de convertir los *scripts* que se ejecutan cuando el paquete se instala o desinstala.

18. Trabajo en grupo en clase

- En grupos de 4 personas...
 - Descargamos un código fuente en formato `tgz`.
 - Tratamos de debianizarlo con `dh_make` y generar el `.deb`
 - Intentad añadirle *scripts* de instalación y desinstalación.
 - Intentad añadirle dependencias.
 - Una vez hecho...convertirlo a `rpm` y `tgz`.
- Echa un vistazo a **`Flatpak`**. ¿En qué consiste? ¿Qué ventajas aporta respecto a formatos como DEB, RPM, etc...?

19. Práctica individual

- Realiza lo mismo que en el trabajo en grupo pero con una práctica tuya de cualquier asignatura.
- Cuando tengas el archivo `".deb"` generado, renómbralo a `"dcapr3.deb"`.

ENTREGA:

- La práctica se entregará en **`pracdlsi`** en las fechas allí indicadas.

20. Aclaraciones

EN NINGÚN CASO ESTAS TRANSPARENCIAS SON LA BIBLIOGRAFÍA DE LA ASIGNATURA.

- Debes estudiar, aclarar y ampliar los conceptos que en ellas encuentres empleando los enlaces web y bibliografía recomendada que puedes consultar en la página web de la **`ficha de la asignatura`** y en la **`web propia de la asignatura`**.