



EBook Gratis

APRENDIZAJE

png

Free unaffiliated eBook created from
Stack Overflow contributors.

#png

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con png.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
libpng versión 1.6.21.....	3
Capítulo 2: Construyendo e instalando tu propio libpng desde la fuente.....	5
Examples.....	5
Simple "cmake" construir e instalar.....	5
Construyendo e instalando libpng con "git" y "configurar".....	5
Capítulo 3: Descargando e instalando libpng desde una distro.....	6
Examples.....	6
Ubuntu, Debian, Mint (sistemas basados en apt).....	6
Fedora, RHEL, CentOS (sistemas basados en yum).....	6
Creditos.....	7

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [png](#)

It is an unofficial and free png ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official png.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con png

Observaciones

Portable Network Graphics es un formato de archivo de gráficos raster que admite la compresión de datos sin pérdida. PNG fue creado como un reemplazo mejorado y no patentado de Graphics Interchange Format (GIF) y es el formato de compresión de imágenes sin pérdida más utilizado en Internet.

PNG admite imágenes basadas en paletas (colores RGB de 24 bits o RGBA de 32 bits), imágenes en escala de grises (con o sin canal alfa) e imágenes RGB [A] a todo color, sin paletas, con un canal alfa opcional. PNG fue diseñado para transferir imágenes en Internet, no para gráficos de impresión de calidad profesional, y no admite espacios de color que no sean RGB, como CMYK.

Los archivos PNG casi siempre usan la extensión de archivo PNG o png y se les asigna imagen / png de tipo de medio MIME. PNG fue aprobado para su uso por el Grupo de Dirección de Ingeniería de Internet el 14 de octubre de 1996, y se publicó como una norma ISO / IEC en 2004.

Para la edición de imágenes, ya sea profesional o de otro tipo, PNG proporciona un formato útil para el almacenamiento de etapas intermedias de edición. La compresión PNG no produce pérdidas y admite truecolor de hasta 48 bits o grayscale de 16 bits. Por lo tanto, guardar, restaurar y volver a guardar una imagen no degradará su calidad a diferencia del JPEG estándar (incluso con la configuración de JPEG de mayor calidad). A diferencia de TIFF, la especificación PNG no deja espacio para que los implementadores elijan y seleccionen qué características admitir; una imagen PNG guardada en una aplicación se puede leer en cualquier otra aplicación compatible con PNG.

Tenga en cuenta que para la transmisión de imágenes truecolor terminadas, especialmente fotográficas, JPEG es casi siempre una mejor opción. La compresión con pérdida de JPEG puede introducir artefactos visibles, pero estos pueden minimizarse y los ahorros en el tamaño del archivo, incluso en niveles de alta calidad, son mucho mejores de lo que generalmente es posible con un formato sin pérdida como PNG. Para imágenes en blanco y negro, en particular de texto o dibujos, la compresión de fax del Grupo 4 de TIFF o el formato JBIG a menudo son mucho mejores que PNG en escala de grises de 1 bit.

La compresión de PNG se encuentra entre las mejores que se pueden obtener sin perder información de la imagen y sin pagar los aranceles de patente. Pero no todas las implementaciones aprovechan al máximo la potencia disponible. Incluso aquellos que lo hacen pueden verse frustrados por elecciones imprudentes por parte del usuario.

Versiones

png

Versión	Fecha de lanzamiento
1.0	1996-10-01
RFC-2083	1997-03-31
1.1	1999-02-15
1.2	1999-07-14
ISO / IEC 15948: 2003	2003-11-10
ISO / IEC 15948: 2004	2004-03-03

Examples

libpng versión 1.6.21

Libpng fue escrito como un complemento de la especificación PNG como una forma de reducir la cantidad de tiempo y esfuerzo que se necesita para admitir el formato de archivo PNG en los programas de aplicación.

Libpng fue diseñado para manejar múltiples sesiones a la vez, para ser fácilmente modificable, para ser portátil a la gran mayoría de las máquinas (ANSI, K&R, 16, 32 y 64 bits) disponibles, y para ser fácil de usar. El objetivo final es promover la aceptación del formato PNG de cualquier manera posible. Si bien aún queda trabajo por hacer (ver el archivo TODO), libpng debería cubrir la mayoría de las necesidades del usuario.

Límites de usuario:

La especificación PNG permite que el ancho y alto de una imagen sea tan grande como 2^{31-1} (0x7fffffff), o aproximadamente 2.147 billones de filas y columnas. Por seguridad, libpng impone un límite predeterminado de 1 millón de filas y columnas. Las imágenes más grandes se rechazarán inmediatamente con una llamada `png_error()`. Libpng puede rechazar imágenes muy amplias debido a posibles condiciones de desbordamiento del búfer, pero puede establecer sus propios límites con:

```
png_set_user_limits(png_ptr, width_max, height_max);
```

Coloque esta declaración después de crear la estructura PNG y antes de llamar a `png_read_info()`, `png_read_png()` o `png_process_data()`.

Al escribir un flujo de datos PNG, ponga esta declaración antes de llamar a `png_write_info()` o `png_write_png()`.

Para recuperar los límites que se están aplicando, use

```
width_max = png_get_user_width_max(png_ptr);
```

```
height_max = png_get_user_height_max(png_ptr);
```

La especificación PNG no establece ningún límite en el número de fragmentos auxiliares permitidos en un flujo de datos PNG. Por defecto, libpng impone un límite de un total de 1000 PLT, tEXt, iTXt, zTXt y trozos desconocidos para almacenar. Si se `info_ptr` tanto `info_ptr` como `end_info_ptr`, el límite se aplica por separado a cada uno. Cambie el límite en el número total de dichos trozos para almacenar con:

```
png_set_chunk_cache_max(png_ptr, user_chunk_cache_max);
```

donde `0x7fffffffL` significa ilimitado. Puede recuperar este límite con:

```
chunk_cache_max = png_get_chunk_cache_max(png_ptr);
```

Libpng impone un límite de 8 Megabytes (8,000,000 bytes) en la cantidad de memoria que una porción comprimida que no sea IDAT puede ocupar cuando se descomprime. Puedes cambiar este límite con:

```
png_set_chunk_malloc_max(png_ptr, user_chunk_malloc_max);
```

y puedes recuperar el límite con:

```
chunk_malloc_max = png_get_chunk_malloc_max(png_ptr);
```

Cualquier parte que cause que se exceda cualquiera de estos límites será ignorada.

Detectando la versión libpng:

La función `png_get_io_ptr()` ha estado presente desde libpng-0.88, nunca ha cambiado y no se ve afectada por las macros de compilación condicional. Es la mejor opción para usar en configurar scripts para detectar la presencia de cualquier versión libpng desde 0.88. En una autoconf "configure.in" usted podría usar

```
AC_CHECK_LIB(png, png_get_io_ptr, ...
```

Lea [Empezando con png en línea](https://riptutorial.com/es/png/topic/5313/empezando-con-png): <https://riptutorial.com/es/png/topic/5313/empezando-con-png>

Capítulo 2: Construyendo e instalando tu propio libpng desde la fuente

Examples

Simple "cmake" construir e instalar

Si desea utilizar "cmake" (consulte www.cmake.org), escriba

```
cd [your libpng source directory]
cmake . -DCMAKE_INSTALL_PREFIX=/path
make
make install
```

donde "/" ruta" apunta al directorio de instalación donde desea colocar los subdirectorios libpng "lib", "include" y "bin".

Construyendo e instalando libpng con "git" y "configurar"

Esto descargará libpng del repositorio oficial "git" y lo construirá en su directorio "libpng".

```
git clone https://github.com/glennrp/libpng.git libpng
cd libpng
./autogen.sh
./configure [--prefix=/path]
make install
```

donde "/" ruta" apunta al directorio de instalación donde desea colocar los subdirectorios libpng "lib", "include" y "bin".

Lea [Construyendo e instalando tu propio libpng desde la fuente en línea](https://riptutorial.com/es/png/topic/6225/construyendo-e-instalando-tu-propio-libpng-desde-la-fuente):

<https://riptutorial.com/es/png/topic/6225/construyendo-e-instalando-tu-propio-libpng-desde-la-fuente>

Capítulo 3: Descargando e instalando libpng desde una distro

Examples

Ubuntu, Debian, Mint (sistemas basados en apt)

```
sudo apt-get install libpng-dev
```

en un sistema x86_64 Ubuntu, esto instala la versión 1.2.x de libpng en

```
libpng12.so.0 => /lib/x86_64-linux-gnu/libpng12.so.0
```

Fedora, RHEL, CentOS (sistemas basados en yum)

```
sudo yum install libpng-devel
```

Lea [Descargando e instalando libpng desde una distro en línea](https://riptutorial.com/es/png/topic/5982/descargando-e-instalando-libpng-desde-una-distro):

<https://riptutorial.com/es/png/topic/5982/descargando-e-instalando-libpng-desde-una-distro>

Creditos

S. No	Capítulos	Contributors
1	Empezando con png	4444 , Community , Dakta Moriamé , Glenn Randers-Pehrson
2	Construyendo e instalando tu propio libpng desde la fuente	Glenn Randers-Pehrson
3	Descargando e instalando libpng desde una distro	Glenn Randers-Pehrson