



EBook Gratis

APRENDIZAJE

SCons

Free unaffiliated eBook created from
Stack Overflow contributors.

#scons

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con SCons.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Empezando.....	2
Capítulo 2: C ++.....	4
Examples.....	4
Una construcción simple.....	4
Especificando varias opciones de compilación.....	4
Capítulo 3: Fases de ejecución de SCons.....	5
Introducción.....	5
Examples.....	5
Inspeccionar fases de SCons.....	5
Capítulo 4: Poner a SCons corriendo.....	6
Introducción.....	6
Examples.....	6
Instalación en Linux.....	6
Instalación en Windows.....	6
Corriendo desde la fuente.....	6
Instalación con Python pip.....	7
Creditos.....	8

Acerca de

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

It is an unofficial and free SCons 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 SCons.

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 SCons

Observaciones

SCons es un sistema de compilación. Toma un montón de archivos de entrada y ejecuta herramientas en ellos para producir salida. SCons está escrito en Python puro, funciona de la misma manera en Linux, Windows y OS X, y puede ejecutarse sin instalación.

Los archivos `SConstruct` SCons son scripts de Python con comandos incorporados que crean un `build tree`. SCons ejecuta el proceso de construcción en fases. Lo primero es leer archivos y construir un árbol de construcción. El segundo es atravesar el árbol para construir archivos de destino.

Versiones

Versión	Fecha de lanzamiento
2.5.1	2017-11-03

Examples

Empezando

Una vez que tenga [SCons en ejecución](#), cree un archivo llamado `SConstruct`:

```
print('..Building World')
```

Ahora ejecuta `scons`:

```
$ scons
scons: Reading SConscript files ...
..Building World
scons: done reading SConscript files.
scons: Building targets ...
scons: `.` is up to date.
scons: done building targets.
```

`SConstruct` es un script de Python con funciones SCons adicionales.

```
Zip('archive', ['SConstruct'])
```

El script anterior se empaqueta en `archive.zip` usando la función `Zip()` provista por SCons. `Zip` es un **Generador**: crea un **objetivo** especificado por el primer argumento de múltiples **fuentes**, que viene como segundo argumento a los Constructores por convención.

SCons **Builders** comienza con una letra mayúscula y opera en el objeto **Environment** , que almacena la configuración de la compilación. SCons proporciona un **entorno** predeterminado, pero puede crearse explícitamente para separar variables de compilación, elegir diferentes herramientas, etc.

```
env = Environment()
env.Zip('archive', ['SConstruct'])
```

Tenga en cuenta que cuando ejecuta el script por segunda vez, no genera nada. SCons reconstruye los objetivos solo cuando los archivos de origen cambian. Modifique `SConstruct` y ejecute `scons` nuevamente para ver la diferencia.

SCons está diseñado para ser extensible. Usted agrega sus propios métodos de Generador adjuntándolos al entorno, que puede tratarse en temas posteriores.

Lea **Empezando con SCons en línea**: <https://riptutorial.com/es/scons/topic/4780/empezando-con-scons>

Capítulo 2: C ++

Examples

Una construcción simple

Es muy fácil construir un proyecto simple de C ++. Aquí hay un ejemplo de un archivo `SConstruct` que lo hace:

```
env=Environment ()

env.Program('hello', Glob('src/*.cpp'))
```

Esto crea el `hello` ejecutable compuesto por todas las fuentes en `src` con la extensión `cpp`.

Especificando varias opciones de compilación

Este ejemplo muestra configuraciones de compilación más detalladas:

```
env=Environment (
    CPPPATH='/usr/include/boost/',
    CPPDEFINES=['foo'],
    LIBS=['bar'],
    SCONS_CXX_STANDARD='c++11')

env.Program('hello', Glob('src/*.cpp'))
```

Esto genera el ejecutable `hello` de todos los archivos `cpp` en `src`, con la siguiente configuración:

- La ruta de búsqueda es `` / usr / include / boost '`
- Se define la constante `FOO`
- Los enlaces ejecutables con `bar`
- C ++ 11 se utiliza como estándar

Lea C ++ en línea: <https://riptutorial.com/es/scons/topic/6158/c-plusplus>

Capítulo 3: Fases de ejecución de SCons

Introducción

SCons es un sistema de compilación de varios pasos. Primero lee todo `SConstruct` y `SConscript` para ejecutar el código Python y crear un `build graph` con objetivos. Luego escanea el sistema de archivos para detectar qué objetivos del `build graph` debe actualizarse, y luego ejecuta el comando para construir objetivos obsoletos.

Examples

Inspeccionar fases de SCons

`scons` describe las fases de ejecución en sí. Al ejecutarlo sobre un `SConstruct` vacío se obtiene esto:

```
$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
scons: `.` is up to date.
scons: done building targets.
```

Para suprimir los mensajes de fase, agregue la opción `-Q`. `--tree=all` permite ver el árbol de dependencias para el objetivo actual que `scons` construyó mientras construía.

```
$ scons -Q --tree=all
scons: `.` is up to date.
+-.
+-SConstruct
```

`.` es el destino predeterminado, que significa "construir `SConstruct` en el directorio actual".

`SConstruct` es una dependencia para construir el objetivo predeterminado.

Lea Fases de ejecución de SCons en línea: <https://riptutorial.com/es/scons/topic/10170/fases-de-ejecucion-de-scons>

Capítulo 4: Poner a SCons corriendo

Introducción

SCons está escrito en Python 2 y no necesita ninguna dependencia para funcionar. Simplemente puede copiar sus scripts al árbol de origen de su proyecto y ejecutar desde aquí. O es posible que desee utilizar la versión empaquetada para su sistema operativo.

Examples

Instalación en Linux

En Debian o Ubuntu, puede instalar SCons usando

```
$ sudo apt-get install scons
```

En sistemas basados en [YUM](#), use

```
$ sudo yum install scons
```

Puede instalar usando un [RPM](#) descargándolo y luego ejecutándolo

```
$ sudo rpm -Uvh http://prdownloads.sourceforge.net/scons/scons-2.5.0-1.noarch.rpm
```

Instalación en Windows

Coge el instalador de <http://scons.org/pages/download.html>

O pruebe la herramienta de instalación de `pip` que viene con Python:

```
pip install scons
```

Si `scons` eso no se pueden encontrar `scons`, asegúrese de que Python `Scripts/` folder se agregue a `PATH` para su instalación de Python.

Corriendo desde la fuente

Si tiene modificaciones para compartir o simplemente quiere probar una nueva versión en desarrollo.

```
$ hg clone https://bitbucket.org/scons/scons
$ python scons/src/script/scons.py
```


Instalación con Python pip

```
pip install scon
```

Si no va a ejecutar `scons` desde la línea de comandos, verifique que el directorio de scripts de Python se haya agregado a PATH para su instalación.

Si desea jugar con la API, `import SCons` de Python no funcionará, porque SCons 2.5.xy a continuación permite instalar varias versiones en paralelo. Esto fue necesario para cambiar entre las diferentes versiones de SCons durante el desarrollo y la solución de problemas. Ahora, la forma más común para esto es usar `virtualenv` o simplemente [ejecutarlo desde la fuente](#) .

Lea [Poner a SCons corriendo en línea](#): <https://riptutorial.com/es/scons/topic/9377/poner-a-scons-corriendo>

Creditos

S. No	Capítulos	Contributors
1	Empezando con SCons	Ami Tavory , anatoly techtonik , bdbaddog , Community
2	C ++	Ami Tavory
3	Fases de ejecución de SCons	anatoly techtonik
4	Poner a SCons corriendo	Ami Tavory , anatoly techtonik