Engineer

Biblioteca

Código empaquetado para que otros programas lo invoquen

Qué es

Una biblioteca es un conjunto de rutinas ya preparadas que un programa utiliza en vez de escribirlas: alguien resolvió una vez un problema, como analizar fechas, comprimir archivos o dibujar un código QR, y publicó la solución con un conjunto de puntos de entrada con nombre que cualquier programa puede utilizar. El plural inglés es libraries. Las bibliotecas C en Linux suelen llevar el prefijo lib-, Windows llama DLL a las compartidas y los programadores de Rust dicen crate, una palabra que el libro de Rust señala que se utiliza de forma intercambiable con biblioteca. La mayoría de los lenguajes también incluyen una biblioteca estándar, los módulos disponibles en cualquier instalación sin añadir nada.

Una biblioteca es un tipo de código reutilizable y resulta más fácil definirla comparándola con conceptos próximos. Una aplicación se ejecuta por sí sola y tiene un usuario; una biblioteca no se ejecuta por sí sola ni tiene un usuario propio; solo se ejecuta dentro de un programa que la llama. La comparación más cercana es con un framework. Con una biblioteca, su código dirige la ejecución y la llama cuando necesita algo; un framework dirige la ejecución y llama a su código en los puntos que define. Por último, un paquete es la caja en la que se distribuye una biblioteca, con un nombre, una versión y una lista de lo que necesita; la biblioteca es el código que hay dentro.

Lo que permite utilizar una biblioteca es su interfaz: las funciones, los tipos y las constantes que se compromete a mantener. El versionado semántico basa todo su esquema en esa promesa. Un proyecto que lo sigue debe declarar una API pública, y su versión mayor aumenta siempre que esa API cambia de forma incompatible.

Program Library HOWTO clasifica las bibliotecas en Linux por el momento en que se incorporan al programa. Una biblioteca estática se copia en el ejecutable cuando se compila. Una biblioteca compartida se carga cuando arranca el programa y la utiliza todo programa que la necesita, por lo que puede actualizarse sin recompilarlos. Una biblioteca de carga dinámica se abre mientras el programa se ejecuta, que es como suelen funcionar los sistemas de complementos.

Utilice el concepto en una decisión

Aplique «¿Qué es una biblioteca?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. La estructura del sistema se define primero sobre el papel para que las piezas encajen y los costes se conozcan antes de escribir código.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Un plan de infraestructura para 14 departamentos — 14 departamentos de investigación con un único plan de infraestructura

Un plan de infraestructura para 14 departamentos Póngase en contacto →

Por qué importa

Cada biblioteca que adopta un proyecto es una decisión con tres respuestas posibles: utilizar una existente, escribir el código internamente o copiar en el repositorio una pieza pequeña y específica y hacerse cargo de ella desde entonces, lo que se denomina vendoring.

Utilizar una biblioteca suele ser lo correcto. Sus autores ya se han encontrado con los casos límite y otros usuarios ya han presentado informes de errores. El coste llega después y es fácil pasarlo por alto: las dependencias de la propia biblioteca llegan con ella, sus avisos de seguridad pasan a ser asunto suyo, su calendario de versiones marca el ritmo de sus actualizaciones y su licencia se incorpora a su producto. Una biblioteca abandonada por su responsable sigue en su compilación hasta que alguien la sustituye.

Escribirla por su cuenta aporta control, pero también exige ocuparse indefinidamente de todo lo que la biblioteca habría ahorrado: los casos límite, las pruebas y el mantenimiento. Es la decisión correcta cuando el problema es específico y central, cuando las bibliotecas disponibles son mucho mayores de lo necesario o cuando el proyecto tiene un motivo para mantener bajo su número de dependencias.

También importa cómo se enlaza una biblioteca. Un programa compilado con una biblioteca estática lleva una copia fija, por lo que una corrección de seguridad de la biblioteca solo llega a los usuarios cuando se vuelve a compilar y distribuir el programa. Una biblioteca compartida puede parchearse una vez para todo el sistema, a cambio de que el programa confíe en la versión que proporcione el sistema. Las licencias no distinguen entre ambas formas de enlace: las preguntas frecuentes de GNU indican que una obra cubierta por la GPL enlazada con otros módulos, de forma estática o dinámica, forma una obra combinada, y los términos de la GPL cubren entonces la combinación completa.

Para quien encarga software, las preguntas prácticas son de qué bibliotecas depende el resultado, bajo qué licencias y quién es dueño de las bibliotecas que aporta el proveedor.

Cómo se aplica en la práctica

Un generador de documentos necesitaba un código QR que apuntara a la versión en línea de cada documento. Cualquiera de las bibliotecas QR existentes habría servido, pero el proyecto se había limitado deliberadamente a dos dependencias en tiempo de ejecución, un motor de plantillas y un renderizador de PDF, para que siguiera instalándose sin problemas mucho después de su publicación. La solución fue incorporar al repositorio un codificador QR de 522 líneas: codificación en modo byte, corrección de errores Reed-Solomon sobre GF(256), patrones de localización, temporización y alineación, información de formato y versión y los ocho patrones de máscara evaluados con las cuatro reglas de penalización. Un decodificador escrito por separado vuelve a leer cada código generado en cada ejecución y comprueba que devuelva la entrada. La página delimita su afirmación: el codificador solo cubre los modos y las versiones que utilizan esos documentos y su razón de ser es el límite de dependencias, no la creencia de que supera a una biblioteca madura.

Un cliente nativo de macOS llama a una biblioteca central mediante su interfaz C, que al principio se representaba con una cabecera escrita a mano. Las llamadas a tres funciones que ya se habían retirado de la biblioteca central se compilaban, se enlazaban y después fallaban en tiempo de ejecución, porque el compilador y el enlazador solo habían visto la representación manual, nunca las declaraciones reales de la biblioteca central. La solución fue un analizador que verifica cada punto de llamada frente a la cabecera real de 7.308 líneas y sus 268 declaraciones, e interrumpe la compilación ante cualquier llamada o constante

de enumeración que la cabecera no declare. Comprueba declaraciones, así que una función que conserva su firma pero cambia su significado sigue pasando la prueba.

Las condiciones de cada encargo establecen a quién pertenece cada cosa. El código escrito para un cliente pasa a ser suyo una vez abonado el importe íntegro. Las bibliotecas y herramientas de propósito general que la empresa escribió antes o para sí misma son la excepción: la propiedad sigue siendo de la empresa y el cliente recibe una licencia que no caduca ni se puede revocar para utilizarlas dentro del trabajo entregado. El código abierto de terceros conserva la licencia elegida por sus autores y se informa al cliente de cuáles de esos componentes se incluyeron en el resultado.

Dónde surgen los problemas

Una actualización rutinaria rompe la compilación

Una dependencia pasa de 2.3 a 2.4 y el código que compilaba ayer falla o se comporta de otro modo. O la biblioteca cambió su interfaz pública sin aumentar la versión mayor, o el proyecto dependía de algo que la biblioteca nunca prometió, como una función interna o un comportamiento no documentado. Fije versiones exactas en un archivo de bloqueo, actualice de forma deliberada consultando el registro de cambios y llame únicamente a lo que la biblioteca documente como público.

El programa confía en una descripción de la biblioteca

Un programa que se compila sin errores termina inesperadamente la primera vez que se utiliza una función. El código se compiló con cabeceras, declaraciones de tipos o enlaces de interfaz (bindings) escritos a mano que se apartaron de la biblioteca realmente enlazada. Genere los enlaces de interfaz a partir de la interfaz publicada de la propia biblioteca o compruebe cada llamada contra ella como parte de la compilación, para que una discrepancia haga fallar la compilación en vez de aparecer ante un usuario.

Una función pequeña arrastra un gran árbol de dependencias

Solo se necesitaba una función y la biblioteca que la proporciona trajo decenas de dependencias propias, cada una con otro conjunto de avisos de seguridad, licencias y actualizaciones. Antes de añadir una biblioteca, examine lo que

arrastra, no solo su propio tamaño. Cuando la necesidad es específica y está bien definida, una pieza pequeña de código incorporado al repositorio con sus propias pruebas puede ser más barata de mantener.

La licencia llega con el código

En una fase avanzada del proyecto, alguien advierte que una biblioteca de un producto de código cerrado tiene una licencia cuyas condiciones se extienden a toda la obra combinada. Nadie comprobó las licencias cuando se añadió la biblioteca, y enlazarla dinámicamente no cambió nada. Compruebe la licencia de cada biblioteca en el momento de añadirla, conserve la lista con la versión publicada y pida a los proveedores sus propias listas.

Lecturas adicionales

- Program Library HOWTO: introducción — qué es una biblioteca de programas en Linux y cómo se distinguen las estáticas, las compartidas y las de carga dinámica.
- El lenguaje de programación Rust: paquetes y crates — crates de biblioteca frente a crates binarios y cómo los contiene un paquete.
- Versionado semántico 2.0.0 — las reglas de numeración de versiones que convierten la API pública de una biblioteca en una promesa.
- La biblioteca estándar de Python — una biblioteca estándar a escala completa: todo lo que se incluye en cada instalación de Python.
- Preguntas frecuentes sobre las licencias de GNU — qué significa la GPL para los programas que enlazan una biblioteca cubierta por ella, de forma estática o dinámica.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es una biblioteca?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/library/