INTRODUCCIÓN A PHP

Maria José Marin Rios

C.E.A.I SENA  TPS2-123

Contenido

[1 ¿Cómo convertir el código PHP en bytecode? 2](#_Toc179366097)

[2 Citar ejemplos de PHP a bytecode 2](#_Toc179366098)

[3 ¿Cómo funciona el compilador JIT? 2](#_Toc179366100)

[4 Investigar y alistar las diferentes extensiones de PHP con su descripción 2](#_Toc179366101)

5. ¿Qué es un servidor web?...................................................................................................2

# Actividad 3.2

# ¿Cómo convertir el código PHP en bytecode?

En PHP, no existe un proceso directo para convertir el código PHP en bytecode, como lo harías en otros lenguajes como Java o Python, que tienen una máquina virtual dedicada para ejecutar bytecode. Sin embargo, PHP sí tiene un mecanismo de optimización a través del uso de OPcache, que compila el código PHP en opcode, un conjunto de instrucciones de bajo nivel que la máquina de PHP (Zend Engine) puede ejecutar directamente, mejorando el rendimiento.

Pasos para convertir el código PHP en opcode usando OPcache:

Activar OPcache: OPcache es una extensión de PHP diseñada para mejorar el rendimiento almacenando en caché el bytecode compilado de los scripts PHP. Para habilitar OPcache, se puede modificar el archivo php.ini.

Abre tu archivo php.ini y asegúrate de que estas líneas estén habilitadas

zend\_extension=opcache.so

opcache.enable=1

opcache.memory\_consumption=128

opcache.interned\_strings\_buffer=8

opcache.max\_accelerated\_files=10000

opcache.revalidate\_freq=2

opcache.fast\_shutdown=1

# Citar unos ejemplos de PHP a Bytecode

Usando bcompiler para generar bytecode persistente bcompiler es una extensión que se usaba en versiones anteriores de PHP para convertir el código PHP en bytecode persistente y almacenarlo en archivos. Sin embargo, es una herramienta obsoleta en las versiones modernas de PHP, pero sigue siendo un buen ejemplo de cómo en el pasado se trabajaba con bytecode.

Ejemplo

php

Copiar código

<?php

$file = fopen("script.phpbc", "wb");

bcompiler\_write\_header($file);

bcompiler\_write\_file($file, "script.php");

bcompiler\_write\_footer($file);

fclose($file);

?>

En este ejemplo, el código PHP se compila en bytecode y se guarda en el archivo script.phpbc. Luego, se puede ejecutar directamente utilizando una versión compatible de PHP que soporte bcompiler.

Fuente: bcompiler en PHP.net

# ¿Cómo funciona el compilador JIT?

El compilador JIT (Just-In-Time) es una técnica de compilación utilizada en lenguajes como PHP (desde la versión 8.0), Java y .NET, cuyo objetivo es mejorar el rendimiento durante la ejecución del código. Funciona compilando partes del código en tiempo de ejecución a código máquina nativo, lo que permite que el sistema lo ejecute directamente, en lugar de depender de una interpretación constante.

En lenguajes que usan compilación JIT, como PHP con Zend Engine, el compilador JIT trabaja de la siguiente manera:

Funcionamiento del JIT en PHP

Interpretación previa: PHP tradicionalmente se ejecuta mediante un intérprete, que convierte el código PHP en opcode (instrucciones de bajo nivel) que el motor Zend puede entender. Normalmente, el opcode se interpreta y ejecuta en tiempo de ejecución.

Compilación JIT: Con la introducción del compilador JIT en PHP 8.0, algunas de las instrucciones del opcode no solo se interpretan, sino que también se compilan a código de máquina nativo en el momento de la ejecución. Esto significa que se convierte directamente en instrucciones del procesador, eliminando la necesidad de interpretación y haciendo que la ejecución sea más rápida.

Optimización en tiempo real: A medida que el código se ejecuta, el compilador JIT puede optimizar dinámicamente secciones del código que detecta que son lentas o críticas para el rendimiento. Esto significa que se realizan optimizaciones sobre la marcha para mejorar aún más el rendimiento.

# Investigar y alistar las diferentes extensiones de PHP con su descripción

1. PDO (PHP Data Objects)

Descripción: PDO proporciona una interfaz uniforme para acceder a diferentes bases de datos en PHP. Permite realizar consultas y manejar transacciones de manera segura, usando declaraciones preparadas para prevenir ataques de inyección SQL.

Uso principal: Acceso a bases de datos (MySQL, PostgreSQL, SQLite, etc.).

Fuente: PHP PDO - Manual oficial

2. mysqli

Descripción: La extensión mysqli es una versión mejorada del antiguo cliente MySQL (mysql). Ofrece soporte para funcionalidades avanzadas como transacciones, consultas preparadas y operaciones en múltiples hilos.

Uso principal: Interacción con bases de datos MySQL.

Fuente: PHP MySQLi - Manual oficial

3. mbstring (Multibyte String)

Descripción: Esta extensión permite el manejo adecuado de cadenas de texto en diferentes codificaciones de caracteres, especialmente las que requieren más de un byte por carácter (multibyte), como UTF-8 o SHIFT-JIS.

Uso principal: Manipulación de cadenas multibyte y soporte para varios juegos de caracteres.

Fuente: PHP mbstring - Manual oficial

4. curl

Descripción: cURL es una extensión que permite realizar peticiones HTTP y otras transferencias de datos a través de una variedad de protocolos (HTTP, FTP, SMTP, etc.). Es muy utilizado para interactuar con APIs externas.

Uso principal: Transferencia de datos y llamadas a APIs RESTful.

Fuente: PHP cURL - Manual oficial

5. gd

Descripción: La extensión GD permite crear y manipular imágenes dinámicamente. Soporta múltiples formatos de imagen, incluidos PNG, JPEG y GIF, y ofrece funciones para crear gráficos y realizar operaciones como redimensionado y rotación de imágenes.

Uso principal: Manipulación y creación de imágenes.

Fuente: PHP GD - Manual oficial

6. intl (Internationalization)

Descripción: intl proporciona herramientas para internacionalización, como la formateación de fechas, números, y la creación de mensajes con traducciones. Utiliza la biblioteca ICU (International Components for Unicode).

Uso principal: Manejo de idiomas y localización.

Fuente: PHP Intl - Manual oficial

7. xml

Descripción: Esta extensión permite trabajar con documentos XML, proporcionando un conjunto de herramientas para analizar, manipular y validar archivos XML.

Uso principal: Procesamiento y manipulación de XML.

Fuente: PHP XML - Manual oficial

8. json

Descripción: La extensión json permite codificar y decodificar datos en formato JSON (JavaScript Object Notation), un formato ligero y ampliamente usado para intercambiar datos entre servidores y aplicaciones web.

Uso principal: Codificación y decodificación de JSON.

Fuente: PHP JSON - Manual oficial

9. openssl

Descripción: Esta extensión permite realizar operaciones de criptografía como el cifrado, la generación de claves y certificados SSL, y la verificación de firmas digitales. Es esencial para la seguridad en aplicaciones PHP.

Uso principal: Cifrado, descifrado y manejo de certificados SSL.

Fuente: PHP OpenSSL - Manual oficial

10. zip

Descripción: Proporciona soporte para la creación y manipulación de archivos comprimidos en formato ZIP. Permite comprimir y extraer archivos directamente desde scripts PHP.

Uso principal: Compresión y descompresión de archivos ZIP.

Fuente: PHP ZIP - Manual oficial

# ¿Qué es un servidor web?

Un servidor web es un sistema que almacena, procesa y entrega contenido web (páginas, imágenes, videos, etc.) a los navegadores de los usuarios a través de Internet o una red local. Cuando un usuario solicita una página web escribiendo una URL en su navegador, el servidor web es responsable de recibir la solicitud, encontrar el contenido apropiado y enviarlo de vuelta al navegador para que se muestre al usuario.

Funciones principales de un servidor web:

Almacenar archivos web (HTML, CSS, JavaScript, imágenes, etc.).

Recibir solicitudes HTTP/HTTPS: El servidor web escucha peticiones de los navegadores en el puerto 80 (HTTP) o 443 (HTTPS).

Procesar solicitudes: Dependiendo de la configuración, puede ejecutar scripts dinámicos como PHP o Node.js.

Responder con contenido: El servidor envía el contenido solicitado (como una página HTML) o un error (404 si no se encuentra la página).

Ejemplos de servidores web comunes:

Apache: Es uno de los servidores más utilizados y es altamente configurable.

Nginx: Conocido por ser rápido y ligero, especialmente en la entrega de contenido estático.

IIS (Internet Information Services): El servidor web de Microsoft.