

# Ganchos de Zsh

Después de explorar “Let Zsh Display Proxy Settings Before Running Network Commands”, profundicé en los hooks de Zsh con ChatGPT. Aquí tienes un resumen conciso para futuras referencias.

---

En Zsh, los hooks te permiten ejecutar funciones personalizadas en puntos específicos durante las operaciones del shell. Además de `preexec`, Zsh ofrece varios hooks para mejorar tu entorno:

## 1. `precmd`

- Cuándo: Antes de que se muestre el prompt.
- Uso: Actualizar el prompt o realizar tareas de limpieza.
- Ejemplo:

```
precmd() {  
    echo "¡Listo para el siguiente comando!"  
}
```

## 2. `chpwd`

- Cuándo: Cuando el directorio actual cambia.
- Uso: Actualizar variables de entorno o activar acciones basadas en el directorio.
- Ejemplo:

```
chpwd() {  
    echo "Cambiado a: $PWD"  
}
```

## 3. `preexec_functions` & `precmd_functions`

- Cuándo: Similar a `preexec` y `precmd`, pero soporta múltiples funciones.
- Uso: Adjunta múltiples acciones sin sobrescribir los hooks existentes.

- Ejemplo:

```
precmd_functions+=(additional_precmd)

additional_precmd() {
    echo "Tarea adicional de precmd."
}
```

#### 4. TRAPDEBUG

- Cuándo: Después de cada comando, antes de mostrar los resultados.
- Uso: Depuración, registro de comandos.
- Ejemplo:

```
TRAPDEBUG() {
    echo "Ejecutado: $1"
}
```

#### 5. TRAPEXIT

- Cuándo: Cuando la shell se cierra.
- Uso: Tareas de limpieza o mostrar mensajes de salida.
- Ejemplo:

```
TRAPEXIT() {
    echo ";Adiós!"
}
```

#### 6. Ganchos de zle

- Cuándo: Durante la edición de línea.
- Uso: Personalizar el comportamiento de la línea de comandos.
- Ejemplo:

```

zle-line-init() {
    echo "Editando un nuevo comando."
}
zle -N zle-line-init

```

## 7. Ganchos de Historial (zshaddhistory, zshremovehistory)

- Cuándo: Al agregar o eliminar entradas del historial.
- Uso: Filtrar o gestionar el historial.
- Ejemplo:

```

zshaddhistory() {
    [[ $1 == *"secret"* ]] && return 1
    return 0
}

```

## 8. periodic

- Cuándo: En intervalos establecidos por period.
- Uso: Verificaciones o actualizaciones rutinarias.
- Ejemplo:

```

periodic() {
    echo "Ejecutando tarea periódica..."
}

```

## 9. add-zsh-hook

- Propósito: Añadir funciones a hooks de manera segura.
- Uso: Añadir múltiples funciones sin sobrescribir las existentes.
- Ejemplo:

```

add-zsh-hook precmd another_precmd

```

```

another_precmd() {

```

```
    echo "Otra función precmd."  
}
```

## Resumen

El sistema de hooks de Zsh es versátil, permitiendo la automatización y personalización:

- `preexec`: Antes de la ejecución de un comando.
- `precmd`: Antes de mostrar el prompt.
- `chpwd`: Al cambiar de directorio.
- `TRAPDEBUG`: Post-comando para depuración.
- `TRAPEXIT`: Al salir del shell.
- Hooks de `zle`: Durante la edición de línea.
- Hooks de Historial: Gestión del historial de comandos.
- `periodic`: En intervalos establecidos.
- `add-zsh-hook`: Agregar múltiples funciones de hook.

Utilizar estos hooks puede mejorar significativamente tu experiencia con Zsh, haciendo que tu shell sea más eficiente y se adapte mejor a tu flujo de trabajo.