

Zsh 鈎子

在探索了「讓 Zsh 在執行網絡命令前顯示代理設置」之後，我與 ChatGPT 一起深入研究了 Zsh 的鈎子 (hooks)。以下是簡要概述，以供將來參考。

在 Zsh 中，鈎子允許你在 shell 操作的特定時刻執行自定義函數。除了 `preexec` 之外，Zsh 還提供了多種鈎子來增強你的環境：

1. `precmd`

- **觸發時機**：在提示符顯示之前。
- **用途**：更新提示符或執行清理操作。
- **示例**：

```
precmd() {  
    echo "準備好執行下一個命令！"  
}
```

2. `chpwd`

- **觸發時機**：當當前目錄改變時。
- **用途**：更新環境變量或根據目錄觸發操作。
- **示例**：

```
chpwd() {  
    echo "已切換到：$PWD"  
}
```

3. `preexec_functions` 和 `precmd_functions`

- **觸發時機**：類似於 `preexec` 和 `precmd`，但支持多個函數。
- **用途**：附加多個操作而不覆蓋現有的鈎子。
- **示例**：

```
precmd_functions+=(additional_precmd)

additional_precmd() {
    echo "額外的 precmd 任務。"
}
```

4. TRAPDEBUG

- **觸發時機**：在每個命令執行後，結果顯示之前。
- **用途**：調試、記錄命令。
- **示例**：

```
TRAPDEBUG() {
    echo "已執行：$1"
}
```

5. TRAPEXIT

- **觸發時機**：當 shell 退出時。
- **用途**：執行清理任務或顯示退出訊息。
- **示例**：

```
TRAPEXIT() {
    echo "再見！"
}
```

6. zle 鉤子

- **觸發時機**：在命令行編輯期間。
- **用途**：自定義命令行行為。
- **示例**：

```
zle-line-init() {
    echo "正在編輯新命令。"
}

zle -N zle-line-init
```

7. 歷史鉤子 (zshaddhistory、zshremovehistory)

- **觸發時機**：添加或刪除歷史記錄時。
- **用途**：過濾或管理歷史記錄。
- **示例**：

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

8. periodic

- **觸發時機**：在 period 設置的時間間隔內。
- **用途**：執行例行檢查或更新。
- **示例**：

```
periodic() {  
    echo "正在執行例行任務..."  
}
```

9. add-zsh-hook

- **目的**：安全地將函數添加到鉤子中。
- **用途**：附加多個函數而不覆蓋現有鉤子。
- **示例**：

```
add-zsh-hook precmd another_precmd  
  
another_precmd() {  
    echo "另一個 precmd 函數。"  
}
```

總結

Zsh 的鉤子系統非常靈活，能夠實現自動化和自定義：

- `preexec`：在命令執行前觸發。
- `precmd`：在提示符顯示前觸發。
- `chpwd`：在目錄改變時觸發。
- `TRAPDEBUG`：在命令執行後用於調試。
- `TRAPEXIT`：在 shell 退出時觸發。
- `zle` 鉤子：在命令行編輯期間觸發。
- 歷史鉤子：管理命令歷史記錄。
- `periodic`：在設定的時間間隔內觸發。
- `add-zsh-hook`：添加多個鉤子函數。

利用這些鉤子可以大大提升你的 Zsh 體驗，使你的 shell 更加高效並符合你的工作流程。