

Система патчингу

install.pl -x проходить по всіх підпапках /usr/local/nodeny/modules і знаходить файли, що починаються з patch або create. Якщо файл починається з patch - вважається, що в ньому є патчі для існуючих файлів. Якщо з create - створення нового файлу.

Закінчення імені файлу не має значення. Рекомендується називати також як той файл, який буде патчитися даним. При цьому якщо файл знаходиться поза папкою web, то в імені файлу необхідно вказати шлях щодо кореня NoDeny, замінюючи зліші на точки.

Наприклад

```
create.test.pl : мається на увазі, що буде створено файл web/_test.pl  
create.kernel.test.txt : kernel/_test.txt  
patch.web.lang.EN.pl : web/lang/_EN.pl
```

Зауважте, що імена результируючих файлів завжди починаються з підкреслення. Вихідні файли ніколи не змінюються! Файли із підкресленнями на початку імені називаються фантомами. Механізм роботи полягає в тому, що якщо NoDeny знаходить фантом - завантажується він, інакше завантажується оригінал.

Така схема не призводить до конфліктів при оновленні - оновлюються оригінали, після чого на них накладаються патчі та записуються у фантоми.

В одному файлі може бути кілька патчів у різні місця файлу або навіть різні файли. Не рекомендується використовувати один файл для патчингу декількох файлів, хоч і допускається, якщо патчі досить дрібні.

Існує 2 методи модифікації файлів: через хуки та через заміни. Хук - спеціальна мітка у файлі, що починається з фрагмента #<HOOK>. Далі йде ім'я хука. Патч є вказівкою замінити хук на такий фрагмент коду.

При оновленні системи, положення хука може змінитися, але це не вплине на патч. він застосовується не за місцем розташування, а на ім'я хука.

Якщо ж у цільовому файлі у потрібному місці немає хука - доведеться скористатися методом заміни. Це не рекомендований спосіб (краще зверніться до автора із проханням створити потрібні хуки).

Формат патчу

```
#<ACTION> file=>'web/lang/RU.pl',hook=>'end'  
$test_test = {  
    msg => 'Yes!!!'  
};
```

Рядок, що починається з #<ACTION>, вказує на початок патчу. У цьому рядку повинен бути параметр file - ім'я файлу, який буде модифікуватися. Якщо застосовується метод хуків, то параметр hook вказує ім'я хука.

У даному прикладі у файлі /usr/local/nodeny/web/lang/RU.pl знайде хук end і в його позицію буде вставлено фрагмент починаючи з 2-го рядка прикладу.

Приклад методу заміन

```
#<ACTION> file=>'web/lang/RU.pl',replace=>' '  
$test_test = {  
    msg => 'Yes!!!'  
};  
#<REPLACE>  
$test_test = {  
    msg => 'No!!!'  
};
```

За цим прикладом буде знайдено фрагмент до #<REPLACE> і замінено фрагмент після #<REPLACE>. Якщо фрагмент не буде знайдений або буде знайдено більше одного збігу - установка модулів перерветься.

--

У деяких випадках патчі потрібно накладати в певному порядку або ж не накладати якщо цільового файлу не існує (припустимо, поточний модуль вносить зміни до інших модулів, деякі з яких можуть бути не встановлені в даній системі).

Застосування патча після патчів модуля dhcp

```
#<ACTION> file=>'web/ajUserIpList.pl',hook=>'show_mac',after=>'dhcp'  
  push @mac_lines, 'Hello';
```

У цьому прикладі, якщо модуль dhcp існує, то будуть застосовані спочатку його патчі до файлу web/ajUserIpList.pl, а потім поточний патч. В іншому випадку просто буде застосований поточний.

Обов'язкова вимога існування модуля xxx

```
#<ACTION> file=>'web/yyy.pl',hook=>'zzz',require=>'xxx'  
  Show 'ok';
```

У параметрі require або after можна вказувати кілька модулів у такому вигляді:

```
require=>['module1', 'module2']
```

From:
<https://ndp.pp.ua/> - **my NoDeny Wiki**

Permanent link:
<https://ndp.pp.ua/doku.php/nodeny/docs/patching>

Last update: **25/01/2023 22:15**

