

Re: Passing hashes to a function

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2004-12/0906.html>

From: Paul Lalli (mritty_at_gmail.com)

Date: 12/10/04

Date: Fri, 10 Dec 2004 20:27:20 GMT

"Shashank Khanvilkar" <shashank@mia.ece.uic.edu> wrote in message
news:cpcu61\$88c\$1@newsx.cc.uic.edu...

> *However, I observe some wierd stuff. In the below program I have two*
> *functions passHashRef_1 and passHashRef_2.*
> *passHashRef_1 modified Hoh(copy of %h1), while passHashRef_2 modifies*
> *copy of %h2.*
> *The original %h2 remains unchanged. But the original %h1 changes.. Why*
> *is that?*
>

Because in the subroutines, you are only doing a shallow copy. That is,
you are only copying the overall hash. The original hash contains
references as values. When you copy the hash, you are copying the
hash's values. That is, you are copying the references. Those
references (both the original and the copies) point to the same data.

```
> #!/usr/bin/perl
>
> my %h1;
>
> $h1{a}{b} = 10;
> $h1{a}{c} = 1;
> $h1{b}{c} = 10;
>
> print "Before: "; print_HoH(%h1);
> passHashRef_1(\%h1, \%h2);
> print "After: "; print_HoH(%h1);
>
> sub passHashRef_1 {
> my ($a, $b) = @_;
> my %aa = %{ $a };
> my %bb = %{ $b };
```

At this point, %aa is a copy of %h1. \$aa{'a'} is a copy of \$h1{'a'}.
However, \$aa{'a'} and \$h1{'a'} are both references, and they both refer
to the same exact data.

```
> $aa{"a"}{"b"} = 100;
```

You are here modifying the value that \$aa{'a'} refers to. You are therefore also modifying the value that \$h1{'a'} refers to.

```
> print "passHashRef_1: "; print_HoH(%aa);  
>  
> }  
>
```

For the canonical solution to this problem, check the FAQ:

```
perldoc -q copy  
"How do I print out or copy a recursive data structure?"
```

Paul Lalli