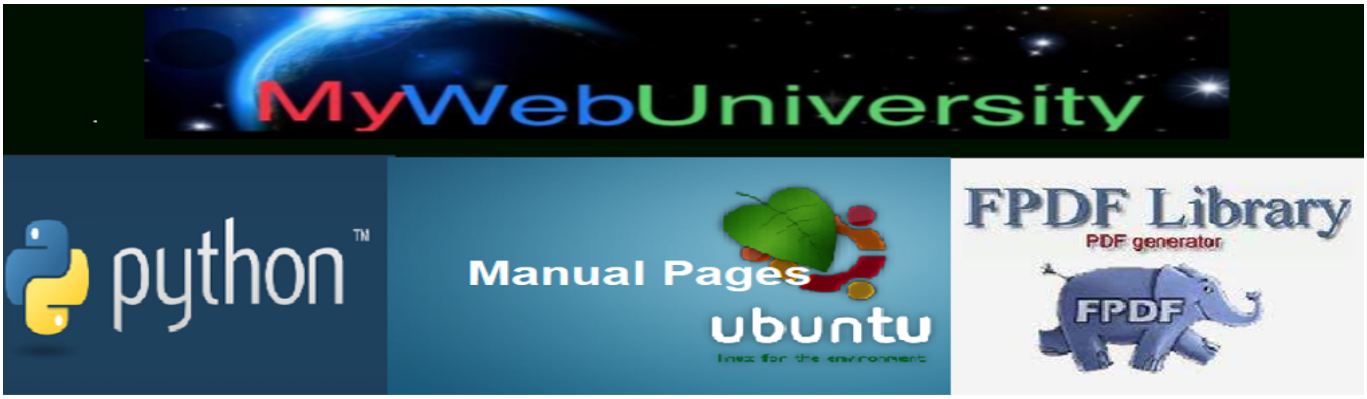

Rocky Enterprise Linux 9.2 Manual Pages on command 'Math::Int128.3pm'

$ man Math::Int128.3pm

Math::Int128(3pm) User Contributed Perl Documentation Math::Int128(3pm)

NAME

Math::Int128 - Manipulate 128 bits integers in Perl

SYNOPSIS

```
use Math::Int128 qw(int128);

my $i = int128(1);
my $j = $i << 100;
my $k = int128("123456789012345678900000");
print($i + $j * 1000000);
```

DESCRIPTION

This module adds support for 128 bit integers, signed and unsigned, to Perl.

In order to compile this module, your compiler must support one of either the "__int128" or "int __attribute__ ((__mode__ (TI)))" types. Both GCC and Clang have supported one or the other type for some time, but they may only do so on 64-bit platforms.

OSX Caveat

On OSX, the system Perl is compiled with both the "-arch x86_64" and "-arch i386" flags. When building this module with a Perl like this, we strip the "-arch i386" flag out,

meaning it is only compiled for the 64-bit architecture. Attempting to use this module while running in 32-bit mode may lead to brokenness. It's also possible that this will cause other problems that we cannot foresee.

Note that if you have built your own non-multiarch Perl on OSX then this will not be an issue.

API

See Math::Int64. This module provides a similar set of functions, just "s/64/128/g" ;-)

Besides that, as object allocation and destruction has been found to be a bottleneck, an alternative set of operations that use their first argument as the output (instead of the return value) is also provided.

They are as follows:

```
int128_inc int128_dec int128_add int128_sub int128_mul int128_pow
int128_div int128_mod int128_divmod int128_and int128_or int128_xor
int128_left int128_right int128_not int128_neg
```

and the corresponding "uint128" versions.

For instance:

```
my $a = int128("129947296068403958476495 3");
my $b = int128("-2849503498690387383748");
my $ret = int128();
int128_mul($ret, $a, $b);
int128_inc($ret, $ret); # $ret = $ret + 1
int128_add($ret, $ret, "12826738463");
say $ret;
```

"int128_divmod" returns both the result of the division and the remainder:

```
my $ret = int128();
my $rem = int128();
int128_divmod($ret, $rem, $a, $b);
```

C API

The module provides a C API that allows to wrap/unwrap int128_t and uint128_t values from other modules written in C/XS.

It is identical to that provided by Math::Int64 so read the documentation there in order to know how to use it.

TODO

Support more operations as log2, pow, etc.

BUGS AND SUPPORT

The C API feature is experimental.

This module requires 128bit integer support from the C compiler. Currently only gcc 4.4 and later are supported. If you have a different compiler that also supports 128bit integers get in touch with me in order to have it supported.

You can send me bug reports by email to the address that appears below or use the CPAN RT bug tracking system available at <http://rt.cpan.org>.

The source for the development version of the module is hosted at GitHub: <https://github.com/salva/p5-Math-Int128>.

My wishlist

If you like this module and you're feeling generous, take a look at my Amazon Wish List: <http://amzn.com/w/1WU1P6IR5QZ42>

SEE ALSO

Math::Int64, Math::GMP, Math::GMPn.

<http://perlmonks.org/?node_id=886488>.

COPYRIGHT AND LICENSE