Mantid at Arch Linux: development at the bleeding edge

Marina Ganeva

JCNS at MLZ, Forschungszentrum Jülich GmbH, Germany

April 2nd 2019

MLZ is a cooperation between

JÜLICH FORSCHUNGSZENTRUM

Helmholtz-Zentrum Geesthacht
Zentrum für Material- und Küstenforschung

Technische Universität München

Arch Linux

Bleeding edge:

> *the most advanced stage of a technology, art, etc., usually experimental and risky (Thesaurus).*

Pro

+ Look to the future: newest versions of libraries, compilers, etcetera
+ Active community
+ Fast bug-fixing

Contra

– Experimental, risky

Arch Linux vs. Ubuntu

	Arch Linux	Ubuntu 18.04
package manager	`pacman`	`apt`
default python	3.7.2	2.7.15
sip	4.19.15	4.19.7
sphinx	1.8.5	1.6.7
numpy	1.16.2	1.13.3
matplotlib	3.0.3	2.1.1
default gcc	8.2.1	7.3.0
boost	1.69	1.65
Qt5	5.12.2	5.9.5

Arch Linux vs. Ubuntu

	Arch Linux	Ubuntu 18.04
default python	24.12.2018	01.05.2018
sip	19.03.2019	23.01.2018
sphinx	10.03.2019	04.02.2018
numpy	26.02.2019	30.09.2017
matplotlib	26.02.2019	10.12.2017
default gcc	26.07.2018	25.01.2018
boost	12.12.2018	07.09.2017
Qt5	15.03.2019	12.04.2018

What's new in Python 3.7

https://docs.python.org/3/whatsnew/3.7.html

Backwards incompatible syntax changes

`async` and `await` are now reserved keywords

What's new in Python 3.7

https://docs.python.org/3/whatsnew/3.7.html

PEP 553: Built-in `breakpoint()` function

Before

```
foo()
import pdb; pdb.set_trace()
bar()
```

After

```
foo()
breakpoint()
bar()
```

Python 3.7: New library modules

`dataclasses`: PEP 557 — Data Classes.

The new `dataclass()` decorator provides a way to declare data classes. A data class describes its attributes using class variable annotations.

```
@dataclass
class Point:
        x: float
        y: float = 0.0
p = Point(1.5)
print(p)
# "Point(x=1.5, y=0.0)"
```

Python 3.7 for Mantid

C-API changes influence Paraview build

`PyUnicode_AsUTF8AndSize()` and `PyUnicode_AsUTF8()` return now type `const char *` rather of `char *`. This concerns:

- `Qt/Python/pqPythonSyntaxHighlighter.cxx`
- `Wrapping/PythonCore/vtkPythonArgs.cxx`

Python 3.7 for Mantid

Module `inspect`

`inspect.getargspec()` is deprecated since Python 3.0. It does not handle function annotations and keyword-only parameters. As a result **IPython does not work** in `MantidPlot`. It is recommended to use `inspect.signature()` or `inspect.getfullargspec()` instead.

Python 3.8: deprecation warnings

Using or importing the ABCs from `collections` instead of from `collections.abc` is deprecated, and in 3.8 it will stop working.

This means that the statement, for example,

```
from collections import Iterable
```

should be replaced with

```
from collections.abc import Iterable
```

`collection.abc` is new since Python 3.3. See more details here `https://docs.python.org/3/library/collections.abc.html`

GCC-8: Stricter rules when using templates

G++ now diagnoses even more cases of ill-formed templates which can never be instantiated, for example code

```
class A { };
template <typename T> struct B {
    bool f() const { return a; }
    A a;
};
```

will produce an error

```
In member function 'bool B<T>::f() const':
error: cannot convert 'const A' to 'bool' in return
bool f() const { return a; }
```

because the type of `B<T>::a` does not depend on `T` and so the function `B<T>::f` is ill-formed

GCC-8: Changes to `alignof` results

The `alignof` operator has been changed to return the minimum alignment required by the target ABI, instead of the preferred alignment (consistent with `_Alignof` in C).

GCC's preferred alignment for standalone variables of type `double` or `long long` is 8 bytes, but the minimum alignment required by the ABI (and so used for non-static data members) is 4 bytes.

Code which uses `alignof` to obtain the preferred alignment can use `__alignof__` instead.

For more information see
`https://gcc.gnu.org/gcc-8/changes.html`

Thank you for your attention!