一般说使用`umcompyle6 *.pyc`命令就可以,但是也会遇到反编译不出来的情况,

core.pyc

这个文件反编译出来的结果是

```
# uncompyle6 version 3.8.0
# Python bytecode 3.7.0 (3394)
# Decompiled from: Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019,
20:34:20) [MSC v.1916 64 bit (AMD64)]
# Embedded file name: certifi\core.py
```

反编译失败了，python3前面16个字节是python头，分别记录了python的版本、编译时间等信息，后面是一个pycodeObject对象，而这个pycodeObject对象中又包含小的pycodeObject对象，所以有可能是其中某一个pycodeObject对象反编译失败而导致整个文件反编译失败，但是如果能够把这些pycodeObject对象，分开反编译就好了，并且能找到到底是哪个pycodeObject对象失败了

部分反编译

我对uncompyle6的代码进行简单封装后，实现了反编译部分pycodeObject对象

`part_reverse.py`

```
import code

from uncompyle6.main import decompile
import sys
# version = (3, 7, 0)
import dis

def get_sub_codeObject_list(co):
    return [ins for ins in list(dis.Bytecode(co)) if "code object" in
str(ins.argval)]

outstream = sys.stdout
showasm = None
showast = False
showgrammar = False
source_encoding = None
mapstream = None
do_fragments = False

from xdis import load_module
filename = "core.pyc"
code_objects = {}
(version, timestamp, magic_int, co, is_pypy, source_size, sip_hash) =
load_module(
    filename, code_objects
```

```
)

def decompile_part(co,father_name=None,outstream=sys.stdout):
    try:
        if father_name is not None:
            name = "%s.%s" % (father_name,co.co_name)
        else:
            name = co.co_name
        outstream.write("\n# %s
_____________________________________________\n" % name)
        decompile(
            version,
            co,
            outstream,
            None,
            False,
            timestamp,
            False,
            None,
            code_objects={},
            source_size=source_size,
            is_pypy=False,
            magic_int=magic_int,
            mapstream=None,
            do_fragments=False,
        )
    except:
        bytecode = get_sub_codeObject_list(co)
        for code in bytecode:
            co = code.argval
            decompile_part(co,name,outstream)
decompile_part(co)
```

执行后结果为

```
# <module> _____________________________________________
# uncompyle6 version 3.8.0
# Python bytecode 3.7.0 (3394)
# Decompiled from: Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019,
20:34:20) [MSC v.1916 64 bit (AMD64)]
# Embedded file name: certifi\core.py
Instruction context:
->
 L.  14        30  LOAD_CONST               None
                32  STORE_GLOBAL             _CACERT_CTX

# <module>.where _____________________________________________
# uncompyle6 version 3.8.0
# Python bytecode 3.7.0 (3394)
```

```
# Decompiled from: Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019,
20:34:20) [MSC v.1916 64 bit (AMD64)]
# Embedded file name: certifi\core.py
if _CACERT_PATH is None:
    _CACERT_CTX = get_path('certifi', 'cacert.pem')
    _CACERT_PATH = str(_CACERT_CTX.__enter__())
return _CACERT_PATH
# global _CACERT_CTX ## Warning: Unused global
# global _CACERT_PATH ## Warning: Unused global
# <module>.read_text ____________________________________________
# uncompyle6 version 3.8.0
# Python bytecode 3.7.0 (3394)
# Decompiled from: Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019,
20:34:20) [MSC v.1916 64 bit (AMD64)]
# Embedded file name: certifi\core.py
with open((where()), 'r', encoding=encoding) as (data):
    return data.read()
# <module>.where ________________________________________________
# uncompyle6 version 3.8.0
# Python bytecode 3.7.0 (3394)
# Decompiled from: Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019,
20:34:20) [MSC v.1916 64 bit (AMD64)]
# Embedded file name: certifi\core.py
f = os.path.dirname(__file__)
return os.path.join(f, 'cacert.pem')
# <module>.contents _____________________________________________
# uncompyle6 version 3.8.0
# Python bytecode 3.7.0 (3394)
# Decompiled from: Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019,
20:34:20) [MSC v.1916 64 bit (AMD64)]
# Embedded file name: certifi\core.py
return read_text('certifi', 'cacert.pem', encoding='ascii')
```

如上所示，主模块是反编译失败的，但是其他子模块都反编译成功了

字节码反编译

现在只要把主模块反编译出来就好了，通过python内置的dis.dis()，可以获取到主模块的字节码指令

`reverse.py`

```
import dis
import marshal
import sys
if len(sys.argv) == 2:
    filename = sys.argv[1]
    with open(filename,"rb") as fp:
        byteCode = fp.read()[16:]

    co  = marshal.loads(byteCode)
    dis.dis(co)
```

命令行输入 `reverse.py core.pyc` ,就得到下面的结果

```
8           0 LOAD_CONST               0 ('\ncertifi.py\n~~~~~~~~~~\n\nThis
module returns the installation location of cacert.pem or its contents.\n')
              2 STORE_NAME               0 (__doc__)

  9           4 LOAD_CONST               1 (0)
              6 LOAD_CONST               2 (None)
              8 IMPORT_NAME              1 (os)
             10 STORE_NAME               1 (os)

 11          12 SETUP_EXCEPT            36 (to 50)

 12          14 LOAD_CONST               1 (0)
             16 LOAD_CONST               3 (('path', 'read_text'))
             18 IMPORT_NAME              2 (importlib.resources)
             20 IMPORT_FROM              3 (path)
             22 STORE_NAME               4 (get_path)
             24 IMPORT_FROM              5 (read_text)
             26 STORE_NAME               5 (read_text)
             28 POP_TOP

 14          30 LOAD_CONST               2 (None)
             32 STORE_GLOBAL             6 (_CACERT_CTX)

 15          34 LOAD_CONST               2 (None)
             36 STORE_GLOBAL             7 (_CACERT_PATH)

 17          38 LOAD_CONST               4 (<code object where at
0x000001D731D08660, file "certifi\core.py", line 17>)
             40 LOAD_CONST               5 ('where')
             42 MAKE_FUNCTION            0
             44 STORE_NAME               8 (where)
             46 POP_BLOCK
             48 JUMP_FORWARD            38 (to 88)

 42     >>   50 DUP_TOP
             52 LOAD_NAME                9 (ImportError)
```

```
             54 COMPARE_OP               10 (exception match)
             56 POP_JUMP_IF_FALSE        86
             58 POP_TOP
             60 POP_TOP
             62 POP_TOP

 47          64 LOAD_CONST               12 (('ascii',))
             66 LOAD_CONST                7 (<code object read_text at
0x000001D731D084B0, file "certifi\core.py", line 47>)
             68 LOAD_CONST                8 ('read_text')
             70 MAKE_FUNCTION             1
             72 STORE_NAME                5 (read_text)

 53          74 LOAD_CONST                9 (<code object where at
0x000001D731D21A50, file "certifi\core.py", line 53>)
             76 LOAD_CONST                5 ('where')
             78 MAKE_FUNCTION             0
             80 STORE_NAME                8 (where)
             82 POP_EXCEPT
             84 JUMP_FORWARD              2 (to 88)
        >>   86 END_FINALLY

 59      >>  88 LOAD_CONST               10 (<code object contents at
0x000001D731D21AE0, file "certifi\core.py", line 59>)
             90 LOAD_CONST               11 ('contents')
             92 MAKE_FUNCTION             0
             94 STORE_NAME               10 (contents)
             96 LOAD_CONST                2 (None)
             98 RETURN_VALUE

Disassembly of <code object where at 0x000001D731D08660, file
"certifi\core.py", line 17>:
 25          0 LOAD_GLOBAL                0 (_CACERT_PATH)
             2 LOAD_CONST                 0 (None)
             4 COMPARE_OP                 8 (is)
             6 POP_JUMP_IF_FALSE         30

 36          8 LOAD_GLOBAL                1 (get_path)
            10 LOAD_CONST                 1 ('certifi')
            12 LOAD_CONST                 2 ('cacert.pem')
            14 CALL_FUNCTION              2
            16 STORE_GLOBAL               2 (_CACERT_CTX)

 37         18 LOAD_GLOBAL                3 (str)
            20 LOAD_GLOBAL                2 (_CACERT_CTX)
            22 LOAD_METHOD                4 (__enter__)
            24 CALL_METHOD                0
            26 CALL_FUNCTION              1
            28 STORE_GLOBAL               0 (_CACERT_PATH)

 39      >> 30 LOAD_GLOBAL                0 (_CACERT_PATH)
            32 RETURN_VALUE
```

```
Disassembly of <code object read_text at 0x000001D731D084B0, file
"certifi\core.py", line 47>:
 48           0 LOAD_GLOBAL              0 (open)
              2 LOAD_GLOBAL              1 (where)
              4 CALL_FUNCTION            0
              6 LOAD_CONST               1 ('r')
              8 LOAD_FAST                2 (encoding)
             10 LOAD_CONST               2 (('encoding',))
             12 CALL_FUNCTION_KW         3
             14 SETUP_WITH              10 (to 26)
             16 STORE_FAST               3 (data)

 49          18 LOAD_FAST                3 (data)
             20 LOAD_METHOD              2 (read)
             22 CALL_METHOD              0
             24 RETURN_VALUE
        >>   26 WITH_CLEANUP_START
             28 WITH_CLEANUP_FINISH
             30 END_FINALLY
             32 LOAD_CONST               0 (None)
             34 RETURN_VALUE

Disassembly of <code object where at 0x000001D731D21A50, file
"certifi\core.py", line 53>:
 54           0 LOAD_GLOBAL              0 (os)
              2 LOAD_ATTR                1 (path)
              4 LOAD_METHOD              2 (dirname)
              6 LOAD_GLOBAL              3 (__file__)
              8 CALL_METHOD              1
             10 STORE_FAST               0 (f)

 56          12 LOAD_GLOBAL              0 (os)
             14 LOAD_ATTR                1 (path)
             16 LOAD_METHOD              4 (join)
             18 LOAD_FAST                0 (f)
             20 LOAD_CONST               1 ('cacert.pem')
             22 CALL_METHOD              2
             24 RETURN_VALUE

Disassembly of <code object contents at 0x000001D731D21AE0, file
"certifi\core.py", line 59>:
 60           0 LOAD_GLOBAL              0 (read_text)
              2 LOAD_CONST               1 ('certifi')
              4 LOAD_CONST               2 ('cacert.pem')
              6 LOAD_CONST               3 ('ascii')
              8 LOAD_CONST               4 (('encoding',))
             10 CALL_FUNCTION_KW         3
             12 RETURN_VALUE
```

参考python文档关于dis模块的介绍可以还原出来主模块的代码

```
__doc__ = '\ncertifi.py\n~~~~~~~~~~\n\nThis module returns the installation
location of cacert.pem or its contents.\n'
import os
try:
    from importlib.resources import path as get_path,read_text
    _CACERT_CTX = None
    _CACERT_PATH = None
    def where():
        pass
except ImportError:
    def read_text(encoding):
        pass
    def where():
        pass
finally:
    def contents():
        pass
```

再把前面部分反编译的结果放进去，就可以得到完整的代码

```
__doc__ = '\ncertifi.py\n~~~~~~~~~~\n\nThis module returns the installation
location of cacert.pem or its contents.\n'
import os
try:
    from importlib.resources import path as get_path,read_text
    _CACERT_CTX = None
    _CACERT_PATH = None
    def where():
        global  _CACERT_PATH
        global _CACERT_CTX
        if _CACERT_PATH is None:
            _CACERT_CTX = get_path('certifi', 'cacert.pem')
            _CACERT_PATH = str(_CACERT_CTX.__enter__())
        return _CACERT_PATH
except ImportError:
    def read_text(encoding):
        with open((where()), 'r', encoding=encoding) as (data):
            return data.read()
    def where():
        f = os.path.dirname(__file__)
        return os.path.join(f, 'cacert.pem')
finally:
    def contents():
        return read_text('certifi', 'cacert.pem', encoding='ascii')
```