Jython

Introduction to Jython programming

Agenda

- Module 1 - Introduction to Jython
- Module 2 - Jython language and semantics
- Module 3 - Data types
- Module 4 - Regular expressions
- Module 5 - Functions, debugging, modules, and packages
- Module 6 - Objects, classes and exceptions
- Module 7 - Java integration
- Module 8 - Testing
- **Module 9 - System programming**
- Module 10 - Conclusion

Topics

- File handling
- Threading
- Queues
- Profiling
- Quiz
- Q & A

Jython

File handling

Common functions

open(fname, access_mode, buffering)
read(byte_cnt) - read the entire file
readline()
readlines() - read a line delimited by a newline
The line separator is defined in
the os module
write(data)
close()

- Module: os
  os.rename()
  os.remove()/os.unlink()

- Module : shutils
  - high level file copy and delete operations

Rename

- Use the os module
  - *rename* renames the file

Delete

- Use the os module
  - *`remove()`* and *`unlink()`* are synonymous

More of os module

Overview

- Miscellaneous OS interfaces
  - Actually a package (os, and os.path)

Function	Description
environ, getenv(name, [, defval]), putenv(name, val)	1. environ is mapping to the environment hash 2. getenv() returns named env variable 3. putenv() creates an environment variable
chdir() getcwd()/curdir()	1. chdir() - change directory 2. getcwd()/curdir() - current working directory
system()	Execute system command
walk()	Walk directory tree (top or bottom)

os.path

- Useful pathname functions

Function	Description
abspath()	Normalized absolute path
basename()	Base name of a path (not the same as the UNIX basename program)
dirname()	Directory name of the path
exists()	Test if path exists
join()	Intelligent building of paths (uses os.sep)
sep	Platform specific path separator
splitext()	Split path into root and extension (leading '.' are ignored

Archive files

Modules

- Three modules
  - gzip - interface to GNU zip and unzip utilities
    - Compression provided by *zlib*
  - zip - interface to create, read, write, append ZIP files
    - Does not support mulit-disk ZIP files
  - tarfile - create a tape archive file
    - Multiple compression methods (depends on what is installed on your system)
- Also - shutil archive methods

Module: shutil

- High level file and file collections copy and delete operations
  - Very fast
- Directory and file operations
- Archiving operations

https://docs.python.org/2/library/shutil.html

Process creation

Module: subprocess

subprocess

- Create subprocesses, connect to their input, output, and error pipes (file descriptors), are obtain the return code
- The subprocess module is meant to replace
  - os.system()
  - os.spawn*
  - os.popen*
  - popen2.*
  - commands.*

Threading

Important

- There is a BIG difference between Python and Jython
  - CPython uses the *Global Interpreter Lock*
    - Only one thread can execute Python code
  - Jython uses Java threads

Jython

Threading

- Interface to manage concurrent threads
  - Builds on thread module
    - Low level primitives - do not use
- Allows programs to to run multiple operations within the same process space

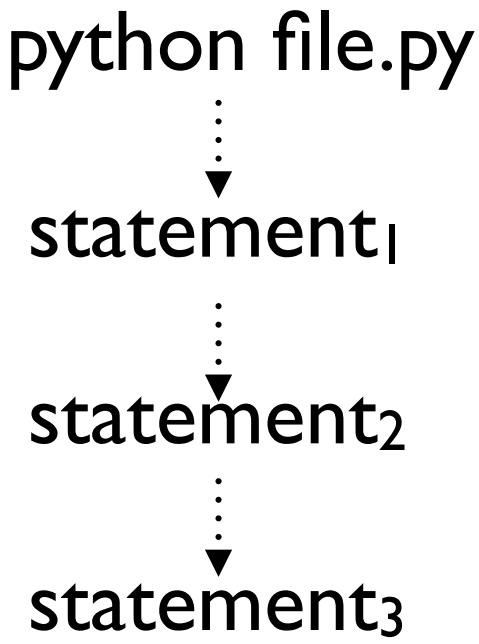
How to do it

- Define function to perform the work
- Create a thread object
  - Always call with keyword arguments

```
import threading
th = threading.Thread(target=worker)
threads.append(th)
th.start()
```

    - *target* argument is the name of the worker function

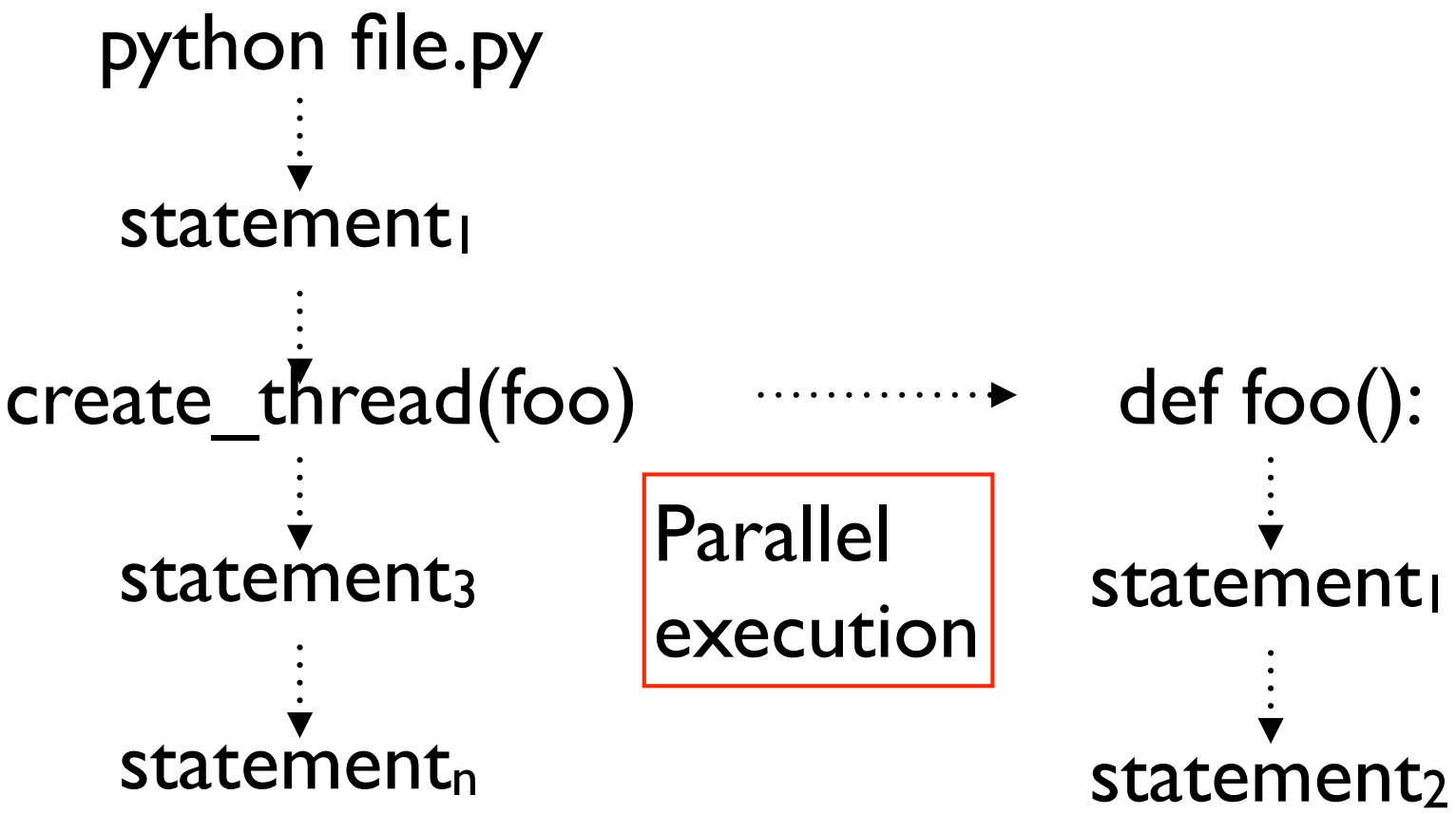
Basics I



At program launch, python begins executing statements.

Main thread

Basics2

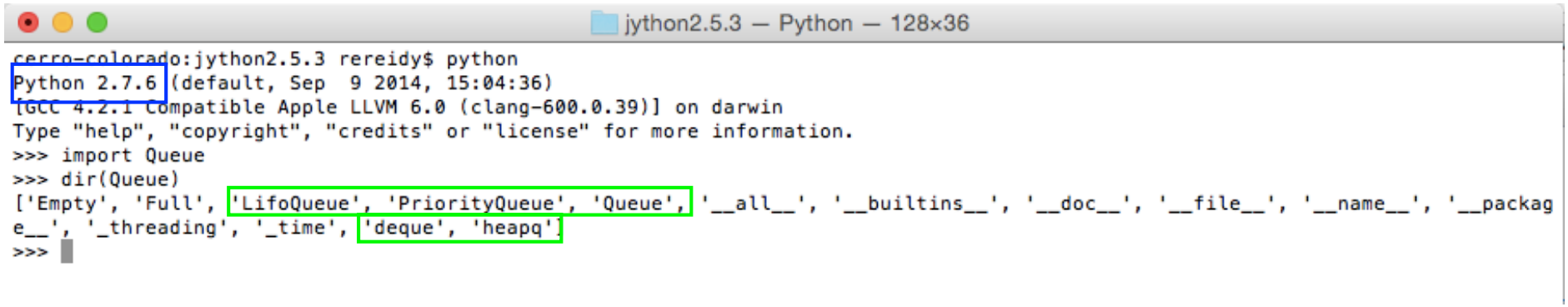


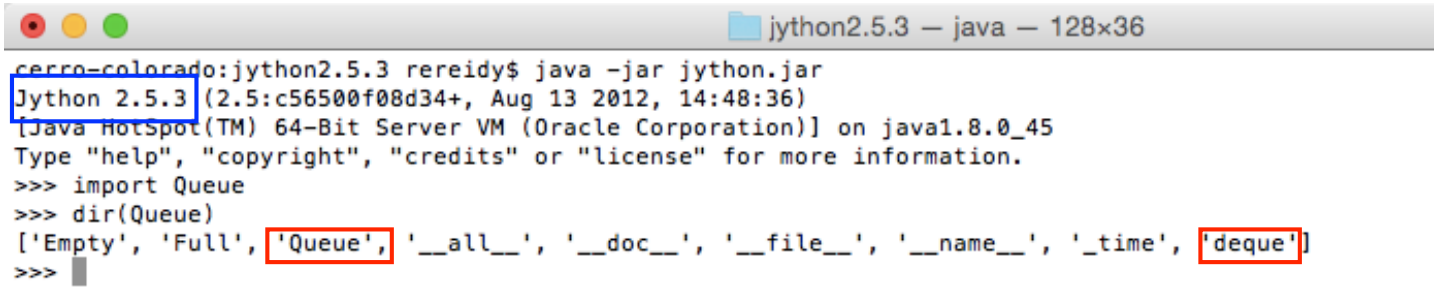
Thread objects

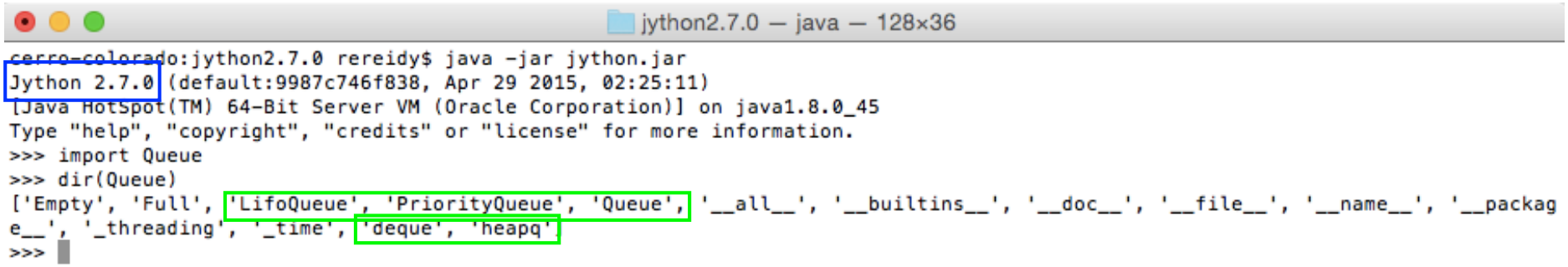
- Separate thread of control
- Once created, just be started by start()
- Other threads can *join()* a thread
- Threads have names (default is "Thread_1")

Queues

- Best practice in Python to use queues
  - Safer programming
    - Resource access







Queues

- FIFO -First in, first out
  - Queue data structure
    - Only Queue in Jython 2.5
- LIFO - Last in, first out
  - Stack data structure
- Priority queue - processing order based on characteristics of items

Profiling

Module: profile

- Collect and analyze statistics for resource consumption by Python programs
- Basic

  run()

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]

         57314 function calls (24 primitive calls) in 1.081 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.001    0.001    0.252    0.252 <string>:0(<module>)
57291/21    0.247    0.000    0.247    0.012 prof1.py:47(fib)
     21/1    0.004    0.000    0.251    0.251 prof1.py:55(fib_seq)
        1    0.830    0.830    1.081    1.081 profile:0(print fib_seq(20); print)
        0    0.000             0.000          profile:0(profiler)
```

Recursive calls

Primitive calls

```
MEMOIZED
========================================================================================
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765]

         103 function calls (45 primitive calls) in 0.013 seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    0.000    0.000    0.006    0.006 <string>:0(<module>)
    59/21    0.003    0.000    0.004    0.000 prof1.py:73(__call__)
       21    0.001    0.000    0.001    0.000 prof1.py:80(bif)
     21/1    0.001    0.000    0.005    0.005 prof1.py:90(bif_seq)
        1    0.007    0.007    0.013    0.013 profile:0(print bif_seq(20); print)
        0    0.000             0.000          profile:0(profiler)
```

Recursive calls

Primitive calls

Quiz

Jython

1. What is the name of the attribute in the os module that defines the platform specific line separator?

A. os.linesep

2. What is the value on your computer?

   A. Mac OS X: '\n'

Jython

3. Are the file contents affected when using shutil.copymode()?

A. No - file contents, owner, and group and not affected.

Jython

4. What is the return value of the system() function?

A. The return value of the spawned (subprocess) command.

5. What does “shell=True” mean in the function calls in the subprocess module?

A. The command supplied will be executed through the shell.

6. What are the security implications of “shell=True”?

A. Unsanitized input from untrusted sources can lead to injection vulnerabilities,

Q & A

Exercises

Jython

1. This will be a two part exercise:
   a. Create a program using shutil to copy a file on your computer.

      After you have copied the file, go out to the OS and make the file unwritable for your account.
   b. Create a 2nd program to copy the permission bits from the first file to the second file.

      Discuss the outcome of number 2 in class.

Jython

2. Create a program which will recursively traverse directories and print a hierarchical file listing:

dirA
- - - - file1.ext
- - - - file2.ext
- - - - dirB
- - - - - - - - - file1.ext
- - - - - - - - - file2.ext
- - - - dirC

etc.

Use os.walk() to generate the file list (see 'help(os.walk)')

3. Create a program to print the file stats for a given file name:

   - size
   - creation time
   - last access time
   - last modified time

   Use jython 2.7 and the collections.namedtuple type to store intermediate results.

   Use os.stat() to get file stats.

Jython

4. Write a program to copy a directory to a temporary directory using shutil. Profile the code and discuss your results.

    a. What at the top 4 reclusive calls?
        i) What is the total CPU time for these calls?
        ii) What modules and methods are being called?
        iii) Examine the module Lib/shutil.py, method copytree(). How does this method get its list of files to copy?