

16. Generic Operating System Services

The modules described in this chapter provide interfaces to operating system features that are available on (almost) all operating systems, such as files and a clock. The interfaces are generally modeled after the Unix or C interfaces, but they are available on most other systems as well. Here's an overview:

- 16.1. `os` — Miscellaneous operating system interfaces
 - 16.1.1. File Names, Command Line Arguments, and Environment Variables
 - 16.1.2. Process Parameters
 - 16.1.3. File Object Creation
 - 16.1.4. File Descriptor Operations
 - 16.1.4.1. Querying the size of a terminal
 - 16.1.4.2. Inheritance of File Descriptors
 - 16.1.5. Files and Directories
 - 16.1.5.1. Linux extended attributes
 - 16.1.6. Process Management
 - 16.1.7. Interface to the scheduler
 - 16.1.8. Miscellaneous System Information
 - 16.1.9. Random numbers
- 16.2. `io` — Core tools for working with streams
 - 16.2.1. Overview
 - 16.2.1.1. Text I/O
 - 16.2.1.2. Binary I/O
 - 16.2.1.3. Raw I/O
 - 16.2.2. High-level Module Interface
 - 16.2.2.1. In-memory streams
 - 16.2.3. Class hierarchy
 - 16.2.3.1. I/O Base Classes
 - 16.2.3.2. Raw File I/O
 - 16.2.3.3. Buffered Streams
 - 16.2.3.4. Text I/O
 - 16.2.4. Performance
 - 16.2.4.1. Binary I/O
 - 16.2.4.2. Text I/O
 - 16.2.4.3. Multi-threading
 - 16.2.4.4. Reentrancy
- 16.3. `time` — Time access and conversions
 - 16.3.1. Functions
 - 16.3.2. Clock ID Constants
 - 16.3.3. Timezone Constants
- 16.4. `argparse` — Parser for command-line options, arguments and sub-commands

- 16.4.1. Example
 - 16.4.1.1. Creating a parser
 - 16.4.1.2. Adding arguments
 - 16.4.1.3. Parsing arguments
- 16.4.2. ArgumentParser objects
 - 16.4.2.1. prog
 - 16.4.2.2. usage
 - 16.4.2.3. description
 - 16.4.2.4. epilog
 - 16.4.2.5. parents
 - 16.4.2.6. formatter_class
 - 16.4.2.7. prefix_chars
 - 16.4.2.8. fromfile_prefix_chars
 - 16.4.2.9. argument_default
 - 16.4.2.10. allow_abbrev
 - 16.4.2.11. conflict_handler
 - 16.4.2.12. add_help
- 16.4.3. The add_argument() method
 - 16.4.3.1. name or flags
 - 16.4.3.2. action
 - 16.4.3.3. nargs
 - 16.4.3.4. const
 - 16.4.3.5. default
 - 16.4.3.6. type
 - 16.4.3.7. choices
 - 16.4.3.8. required
 - 16.4.3.9. help
 - 16.4.3.10. metavar
 - 16.4.3.11. dest
 - 16.4.3.12. Action classes
- 16.4.4. The parse_args() method
 - 16.4.4.1. Option value syntax
 - 16.4.4.2. Invalid arguments
 - 16.4.4.3. Arguments containing -
 - 16.4.4.4. Argument abbreviations (prefix matching)
 - 16.4.4.5. Beyond sys.argv
 - 16.4.4.6. The Namespace object
- 16.4.5. Other utilities
 - 16.4.5.1. Sub-commands
 - 16.4.5.2. FileType objects
 - 16.4.5.3. Argument groups
 - 16.4.5.4. Mutual exclusion
 - 16.4.5.5. Parser defaults
 - 16.4.5.6. Printing help
 - 16.4.5.7. Partial parsing

- 16.4.5.8. Customizing file parsing
 - 16.4.5.9. Exiting methods
 - 16.4.6. Upgrading optparse code
- 16.5. getopt — C-style parser for command line options
- 16.6. logging — Logging facility for Python
 - 16.6.1. Logger Objects
 - 16.6.2. Logging Levels
 - 16.6.3. Handler Objects
 - 16.6.4. Formatter Objects
 - 16.6.5. Filter Objects
 - 16.6.6. LogRecord Objects
 - 16.6.7. LogRecord attributes
 - 16.6.8. LoggerAdapter Objects
 - 16.6.9. Thread Safety
 - 16.6.10. Module-Level Functions
 - 16.6.11. Module-Level Attributes
 - 16.6.12. Integration with the warnings module
- 16.7. logging.config — Logging configuration
 - 16.7.1. Configuration functions
 - 16.7.2. Configuration dictionary schema
 - 16.7.2.1. Dictionary Schema Details
 - 16.7.2.2. Incremental Configuration
 - 16.7.2.3. Object connections
 - 16.7.2.4. User-defined objects
 - 16.7.2.5. Access to external objects
 - 16.7.2.6. Access to internal objects
 - 16.7.2.7. Import resolution and custom importers
 - 16.7.3. Configuration file format
- 16.8. logging.handlers — Logging handlers
 - 16.8.1. StreamHandler
 - 16.8.2. FileHandler
 - 16.8.3. NullHandler
 - 16.8.4. WatchedFileHandler
 - 16.8.5. BaseRotatingHandler
 - 16.8.6. RotatingFileHandler
 - 16.8.7. TimedRotatingFileHandler
 - 16.8.8. SocketHandler
 - 16.8.9. DatagramHandler
 - 16.8.10. SysLogHandler
 - 16.8.11. NTEventLogHandler
 - 16.8.12. SMTPHandler
 - 16.8.13. MemoryHandler
 - 16.8.14. HTTPHandler
 - 16.8.15. QueueHandler
 - 16.8.16. QueueListener

- 16.9. `getpass` — Portable password input
- 16.10. `curses` — Terminal handling for character-cell displays
 - 16.10.1. Functions
 - 16.10.2. Window Objects
 - 16.10.3. Constants
- 16.11. `curses.textpad` — Text input widget for curses programs
 - 16.11.1. Textbox objects
- 16.12. `curses.ascii` — Utilities for ASCII characters
- 16.13. `curses.panel` — A panel stack extension for curses
 - 16.13.1. Functions
 - 16.13.2. Panel Objects
- 16.14. `platform` — Access to underlying platform's identifying data
 - 16.14.1. Cross Platform
 - 16.14.2. Java Platform
 - 16.14.3. Windows Platform
 - 16.14.3.1. Win95/98 specific
 - 16.14.4. Mac OS Platform
 - 16.14.5. Unix Platforms
- 16.15. `errno` — Standard `errno` system symbols
- 16.16. `ctypes` — A foreign function library for Python
 - 16.16.1. `ctypes` tutorial
 - 16.16.1.1. Loading dynamic link libraries
 - 16.16.1.2. Accessing functions from loaded dlls
 - 16.16.1.3. Calling functions
 - 16.16.1.4. Fundamental data types
 - 16.16.1.5. Calling functions, continued
 - 16.16.1.6. Calling functions with your own custom data types
 - 16.16.1.7. Specifying the required argument types (function prototypes)
 - 16.16.1.8. Return types
 - 16.16.1.9. Passing pointers (or: passing parameters by reference)
 - 16.16.1.10. Structures and unions
 - 16.16.1.11. Structure/union alignment and byte order
 - 16.16.1.12. Bit fields in structures and unions
 - 16.16.1.13. Arrays
 - 16.16.1.14. Pointers
 - 16.16.1.15. Type conversions
 - 16.16.1.16. Incomplete Types
 - 16.16.1.17. Callback functions
 - 16.16.1.18. Accessing values exported from dlls
 - 16.16.1.19. Surprises
 - 16.16.1.20. Variable-sized data types
 - 16.16.2. `ctypes` reference
 - 16.16.2.1. Finding shared libraries
 - 16.16.2.2. Loading shared libraries

- 16.16.2.3. Foreign functions
- 16.16.2.4. Function prototypes
- 16.16.2.5. Utility functions
- 16.16.2.6. Data types
- 16.16.2.7. Fundamental data types
- 16.16.2.8. Structured data types
- 16.16.2.9. Arrays and pointers