

[index](#)**client** [c:\users\finn\csci351\hw3\client.py](#)

module that sends a file to a server

Modules

[argparse](#) [rdt](#)
[random](#) [socket](#) [time](#)

Functions

Lock = `allocate_lock(...)`
`allocate_lock()` -> lock object
(`allocate()` is an obsolete synonym)

Create a new lock object. See `help(type(threading.Lock()))` for information about locks.

excepthook = `_excepthook(...)`
[excepthook](#)(`exc_type`, `exc_value`, `exc_traceback`, `thread`)

Handle uncaught `Thread.run()` exception.

get_ident(...)
[get_ident\(\)](#) -> integer

Return a non-zero integer that uniquely identifies the current thread amongst other threads that exist simultaneously.
This may be used to identify per-thread resources.
Even though on some platforms threads identities may appear to be allocated consecutive numbers starting at 1, this behavior should not be relied upon, and the number should be seen purely as a magic cookie.
A thread's identity may be reused for another thread after it exits.

get_native_id(...)
[get_native_id\(\)](#) -> integer

Return a non-negative integer identifying the thread as reported by the OS (kernel). This may be used to uniquely identify a particular thread within a system.

main()
main function that parses file argument and sends the file packets to the server

read_file(path)
reads in a file and splits it into packets
path: filepath to read in

stack_size(...)
[stack_size](#)([`size`]) -> size

Return the thread stack size used when creating new threads. The optional size argument specifies the stack size (in bytes) to be used for subsequently created threads, and must be 0 (use platform or configured default) or a positive integer value of at least 32,768 (32k). If changing the thread stack size is unsupported, a ThreadError exception is raised. If the specified size is invalid, a ValueError exception is raised, and the stack size is unmodified. 32k bytes currently the minimum supported stack size value to guarantee sufficient stack space for the interpreter itself.

Note that some platforms may have particular restrictions on values for the stack size, such as requiring a minimum stack size larger than 32 KiB or requiring allocation in multiples of the system memory page size - platform documentation should be referred to for more information (4 KiB pages are common; using multiples of 4096 for the stack size is the suggested approach in the absence of more specific information).

wait_for_ack()

runs on a separate thread and listens for ack response packets, increases the base of the window for sending packets

Data

```

TIMEOUT_MAX = 4294967.0
acked_packets = {}
base = 0
client_port = 8050
lock = <unlocked _thread.lock object>
next_seq = 0
packets = []
s = <socket.socket fd=412, family=2, type=2, proto=0, laddr=('127.0.0.1', 8050)>
server_addr = ('127.0.0.1', 8000)
server_port = 8000
timeout = 5
timestamp = {}
wait = 5
window_size = 3

```