## Main greenlet: main gevent.spawn(foo) Greenlet.spawn(foo) fo = Greenlet(foo) greenlet. init (None, get hub()) get hub() Hub() greenlet. init () loop, loop\_class = 'gevent.core.loop' => 'gevent.libev.corecffi.loop' run = foo fo.start() \_start\_event = hub.loop.run\_callback(fo.switch) gevent.spawn(bar) gevent.joinall([greenlets]) wait(greenlets): wait for all greenlets to become ready list(iwait(greenlets)): iteratively yield greenlets as they are ready waiter MultipleWaiter for g in greenlets: g.rawlink(waiter.switch) for in xrange(len(greenlets)): item = waiter.get() waiter.clear() yield item

#### hub

parent?
loop = libev loop
switch()
wait(watcher)
run()
while True:
loop.run()

#### foo greenlet: fo

parent = hub run = foo start event = cb(fo.switch) links = [mw.switch] start() hub.loop.run callback(fo.switch) switch() rawlink(waiter.switch) links.append(waiter.switch) report result(res) hub.loop.run callback( links) run()  $res = _run()$ report result(res) foo(): print(...) gevent.sleep(0) waiter Waiter hub.loop.run callback(fow.switch) waiter.get() print(...)

### bar greenlet: ba

parent = hub

\_run = bar
start()
switch()
rawlink()
bar():
print(...)
gevent.sleep(0)
print(...)

## \_MultipleWaiter: mw

\_values = []
switch(value)
 \_values.append(value)
 Waiter.switch()
get() => call from greenlets
 Waiter.get()
 Waiter.clear()

#### Waiter

```
hub = hub
greenlet = main

__init__()
hub = hub
switch() => call from hub only
main.switch()
get() => call from greenlets
greenlet = getcurrent() => main
return hub.switch()
clear()
greenlet = None
```

## libev loop: libev.corecffi.loop

\_callbacks
 [cb(fo.switch), ba, fow, baw, mw, mw]
 \_prepare
 \_run\_callbacks()
 \_ptr = libev.gevent\_ev\_default\_loop()

run\_callback(func)
 \_callbacks.append(callback(func))
 return cb(func)
 \_run\_callbacks()
 for cb in \_callbacks:
 cb() => fo.switch()

run()
 libev.ev\_run(self.\_ptr, flags)
 \_prepare run
 \_run\_callbacks()

#### waiter: fow

hub = hub
greenlet = fo

switch() => call from hub only
 fo.switch()
get() => call from greenlets
 greenlet = getcurrent() => fo
 return hub.switch()

# waiter: baw

hub = hub greenlet = ba switch() ba.switch() get() greenlet = ba hub.switch()