

Forum:[Hack Forums](#)

Thread: [My script wont run on large text files](#)

41phafr34k - March 15, 2017, 11:32 pm

I wrote a code to parse out words with less then 8 characters and take out duplicates from wordlists. I made few small test files while i wrote it and it worked perfect by the time I finished (I'm no coder... yet... so this small script took me a while).

Anyways the problem I am having now is that I tried to run some larger files through it and it wont even fix the first 10 lines.

Is this something to do with my code or is it due to the file sizes, should I just do it in bash? I wanted to use python because thats what I am trying to teach myself and thought it would be a great project. Anyways, here is the code....

```
Code #!/usr/bin/env python2
import os
```

```
def parser():
    f = open(file)
    output = []
    for line in f:
        if len(line) >= 9:
            output.append(line)
    f.close()
    f = open(file, 'w')
    f.writelines(output)
    f.close()
```

```
def del_dupes():
    os.system('sort -u "%s" > "%s"'%(file, new_file))
```

```
print """Ok, so let's get started...
This file will parse out all words
with less then 8 letters and also
delete duplicates so you may you
use your wordlists for proper WPA
cracking. At the end you will have
a new file ready for WPA cracking."""
```

```
file = raw_input("What is the name of the file to parse?>>> ")
new_file = 'new_'+file
```

```
parser()
del_dupes()
```

```
print "We're all done..."
print "You're new file name is %s" % new_file
I'm going to try to find out at what point (size) it begins giving issues.
```

I've narrowed down the problem to specific wordlists I made a few of my own by just making 100

lineas then copying them over and over and over again then selecting all copy and repeating as necessary to get to 22.7MB which is the smallest size I had trouble with and it worked fine.

But still if anyone see's an issue with how I am going about this, please let me know.

Folks - March 16, 2017, 12:47 am

Can you show what one of your text files looks like?

41phafr34k - March 16, 2017, 1:37 am

Code 0-1-laws

0-recurrent

0-summable

00000000

0-1-laws

0-recurrent

0-summable

00000000

00brucellosis

00kiribati

00murree

0-1-laws

0-recurrent

0-summable

00000000

00brucellosis

g
g
g
g
g
g
g
g
g
g
g
g
g
g

g
g
g
g
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg

g
g
g
g
g
g
g
g
g
g
g
g
g
g
g
g
g

gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg
gggggggggggggggggggg

g
g
g
g
g
g
g
g
g
g
g

g

[illegible]

00kiribati

00murree

00whitebait

01-matrices

01-matrix

01-sequences

01-valued

01-vector

02nnsylvania

02ovakia

02ppajarvi

04bashing

04hopper

0abalone

0abandon

0abcdefg

0abdomen

0abetted

0abeyant

0abolish

0abreact

0abreast

0abridge

0abscess

0absence

0absolve

0abstain

0abusive

0abutted

0abysmal

0academy

0acclaim

0account

0accrual

0acetate

0acetone

0achieve

0acolyte

0acquire

0acreage

0acrobat

0acronym

0acrylic

0actinic

0actress

0actuate

0acyclic

0adamant

0addenda

0address

0adenoma

0adjoint

0adjourn

0adjudge

0adjunct

0admiral

0adrenal

0adulate

0advance

0adverse

0advisee

0advisor

0aeolian

0aerobic

0aerosol

0affable

0afflict

0affront

0against

0agitate

0agnomen

0aileron

0airdrop

0airfare

0airflow

0airlift

0airline

0airlock

0airmail

0airmass

0airpark

0airport

0albumin

0alchemy

0alcohol

0alewife

0alfalfa

Oalfonso

Oalgebra

Oaliases

Oalimony

Oaliquot

Oallegro

Oallergy

Oalmanac

Oalready

Oalumina

Oalumnae

Oalumnus

Oalundum

Oalveoli

Oalyssum

Oamadeus

Oamalgam

Oamanita

Oamateur

Oamatory

Oambient

Oambling

Oambrose

Oammeter

Oammonia

Oamoebae

Oamongst

Oamplify

Oanagram

Oanalogy

Oanalyst

Oanarchy

Oanatomy

Oanchovy

Oancient

Oanemone

Oangelic

Oangling

Oanguish

Oangular

Oaniline

Oanimals

Oanimate

Oanimism

Oanionic

Oannuity

Oannular

Oannulus

Oanomaly

Oanother

Oantacid

Oantenna

Oantigen

0antique
0anxiety
0anxious
0anybody
0apatite
0aphasia
0aphasic
0apology
0apostle
0apparel
0appease
0applaud
0applied
0appoint
0apprise
0approve
0apricot
0apropos
0aquatic
0aqueous
0arbiter
0arbutus
0archaic
0archery
0arching
0archive

0arcsine
0ardency
0arduous
0areaway
0ariadne
0armhole
0armload
0armoire
0arousal
0arraign
0arrange
0arrival
0arsenal
0arsenic
0article
0artisan
0artwork
0ascetic
0ascribe
0aseptic
0ashtray
0asinine
0askance
0asocial
0asphalt
0aspirin
0assault

0asshole

0assuage

0asteria

0astound

0astride

0asunder

0atavism

0atheist

0athlete

0athwart

0atrophy

0attache

0attempt

0attract

0auction

0audible

0auditor

0augment

0austere

0automat

0autopsy

0avarice

0average

0averred

0avionic

0-1-laws

0-1-laws

0-1-laws

0-recurrent

0-summable

00000000

00brucellosis

00kiribati

00murree

00whitebait

01-matrices

01-matrix

0-1-laws

0-recurrent

0-summable

00000000

00brucellosis

00kiribati

00murree

00whitebait

01-matrices

01-matrix

01-sequences

01-valued

01-vector

02nnsylvania

0-1-laws

0-1-laws

0-recurrent

0-summable

00000000

00brucellosis

00kiribati

00mthe 'G's and 'g's were just me seeing if I added stuff into the list and see if they'd work and they didn't

thats not an entire lists just what I added into a premade list.

a test list of my mine is:

Code 00000000

00106i57

00106ic41

00106ic4119

00106iz3

004n6ium

005736i73

005736i7ic

005p0120u5

005p0123

005p012343

005p0124n63

005p0124n6ium

005p012ic

005p012if3120u5

005p312m

005ph3123

0060ni0ph0123

0060ni41

0060nium

006124ph

0063n35i5

0063n37ic

0064m0u5

0064m373

0070c0id

0070c0id34

0070c0id34n

0070c0u5

007h3c41

00b1457ic

00c935i5

00c9574c30u5

00c9574c343

00c957i5

00c957ic

00fbi12d

00kin35i5

00kin373

00kin37ic

00m37129

00m3712ic

00m37312

00m4n7i4
00m9c370u5
00m9c373
00m9c3735
00p012ph912in
00p145mic
00ph01201212h4ph9
00ph01203pi13p59
00ph0120541pin63c70m9
00ph0120570m9
00ph012070m9
00ph0120m4
00ph0120m414ci4
00ph0120m4ni4
00ph0120n
00ph0120p3x9
00ph0123
00ph01230c313
00ph0123c70m9
00ph01\$100-a-plate dinner
10 Downing Street
10-point
11-point
1200 hours
12-point
16-point
18-point
20-point
2,4,5-t
3.14159
3.141592
3-D display
48-point
5-poin

I also tried taking traits from the bad lists and putting them in my good ones and always the same. I tried playing with spacing since my original test lists had no empty lines. tried playing with different non essential chars like \$% or /n and things like that effecting python code and no difference. My own lists worked and pre-made did not.

I know one of them may have got corrupted by improperly stopping a premature torrent download and another has been subject to multiple compressions/decompressions so those may have played a role in it all. I just wanted to make sure before I go ahead and stash this script away and start a new one.

The end result will be a program but I am breaking it up into sections and small scripts until I am ready to put it altogether and debug.

Folks - March 16, 2017, 12:52 pm

So I am not the best programmer but this is what I have that works. It doesn't remove special characters or spaces but that's not hard to add.

PHP Code
<?php

```
def parser(filename):
    temp = []
    with open (filename, "r") as f:
        temp = f.readlines()
    for line in temp:
        if len(line) < 9:
            temp.remove(line)
    return temp
```

```
def noDuples(seq, idfun=None):
    if idfun is None:
        def idfun(x): return x
    seen = {}
    result = []
    for item in seq:
        marker = idfun(item)
        if marker in seen: continue
        seen[marker] = 1
        result.append(item)
    return result
```

```
filename = "t.txt"
print(*noDuples(parser(filename)))
```

You could also try using readline instead of readlines to read each line individually.

noDuples taken from f5 [here](#).

41phafr34k - March 16, 2017, 1:06 pm

Thanks I just woke up but Ill give that code a try and see what happens. I did some slight modifications to my code last night myself and I am still not seeing a difference between my 22MB list made by yours truly and the 22MB list I download, both are .txt files...

But I'll run a series of tests over all the options on the table now and see if I get different results

SIM-1866423 - March 16, 2017, 1:10 pm

Try fileinput library to prevent overloading.

41phafr34k - March 16, 2017, 2:32 pm

Im trying to learn file input here since I could not get Folks script to work for me (even with some minor changes), but everytime i try to impliment fileinput i run into errors fix one and 2 more pop up. I guess thats programming but could someone assist me in giving me a hint at least on how to convert my initial parser function to fileinput then I can take it from there with the second function.

Code `#!/usr/bin/env python2`

```
import os
import sys
import fileinput
```

```
def parser():
    f = fileinput.input(file)
    parsed = open('parsed_'+file, 'w')
    sys.stdout = parsed
```

```

output = []
for line in f:
    if len(line) >= 9:
        output.append(line)
        sys.stdout.write(line)

def del_dupes():
    f2 = 'parsed_'+file
    cleanfile = 'cleaned_'+file
    os.system('sort -u "%s" > "%s"'%(f2, cleanfile))

```

print ""Ok, so let's get started...
This file will parse out all words
with less than 8 letters and also
delete duplicates so you may use
your wordlists for proper WPA
cracking. At the end you will have
a new file ready for WPA cracking.""

```

file = raw_input("What is the name of the file to parse?>>> ")
parser()
del_dupes()

```

print "We're all done..."This is what I have come up with which works great when I ran it using only
the parser function but when I add in del_dupes the final cleanfile saves as an empty file yet the
parser file is still good. Any ideas?

41phafr34k - March 17, 2017, 8:54 pm

At the risk of looking like I gave up I ended off with the following:

Code `#!/usr/bin/env python2`

```

import os
import sys
import fileinput

```

```

def parser():
    f = fileinput.input(file)
    parsed = open('parsed_'+file, 'w')
    sys.stdout = parsed
    for line in f:
        if len(line) >= 9:
            sys.stdout.write(line)

```

```

def del_dupes():
    f2 = fileinput.input(file2)
    cleaned = open('cleaned_'+file, 'w')
    sys.stdout = cleaned
    nodupes = set()
    for line in f2:
        if line in nodupes:
            pass
        else:

```

```
nodupes.add(line)
sys.stdout.write(line)
```

```
print ""Ok, so let's get started...
This file will parse out all words
with less then 8 letters and also
delete duplicates so you may you
use your wordlists for proper WPA
cracking. At the end you will have
a new file ready for WPA cracking.""
```

```
file = raw_input("What is the name of the file to parse?>>> ")
file2 = 'parsed_'+file
```

```
parser()
del_dupes()
```

So far it has worked very very well. Have decided to give up after hitting 1GB file but nonetheless I am fine with that because files to large end up causing problems in other ways during cracking so I tend to split them up.

Las thing to do erase the file created in between the original file and the final file but I am kinda busy working on some other scripts, twice as advanced that will take much longer, so I will come back to it. I'm sure it will be something just as simple as running file.remove at the end or something but again, I'm busy and dont wanna get stuck working on it if I am wrong.

.kr. - March 18, 2017, 1:17 pm

just use

```
Code $ strings -n 9 [file] | sort | uniq
its much faster and on the most linux machines available
```

41phafr34k - March 20, 2017, 1:22 pm

^^You're right about that...

However, although I do need a script that does this for me, I can also find one premade on the net. I would rather do the entire script in python without passing bash commands into it. This is just a small practice project for myself to strengthen my noob python skills. Wont learn unless I actually write stuff.

But yeah you are right.

Here is the latest and final version since I've moved onto bigger and better things, plus I started my sys admin course so I'm too busy to do hours and hours a day of coding.?

```
Code #!/usr/bin/env python2
import os
import sys
import fileinput
```

```
def parser():
    f = fileinput.input(file)
    parsed = open('parsed_'+file, 'w')
```

```

sys.stdout = parsed
for line in f:
    if len(line) >= 9:
        sys.stdout.write(line)

def del_dupes():
    f2 = fileinput.input(file2)
    cleaned = open('cleaned_'+file, 'w')
    sys.stdout = cleaned
    nodupes = set()
    for line in f2:
        if line in nodupes:
            pass
        else:
            nodupes.add(line)
            sys.stdout.write(line)

```

print ""Ok, so let's get started...
This file will parse out all words
with less than 8 letters and also
delete duplicates so you may use
your wordlists for proper WPA
cracking. At the end you will have
a new file ready for WPA cracking.""

```

file = raw_input("What is the name of the file to parse?>>> ")
file2 = 'parsed_'+file

```

```

parser()
del_dupes()

```

.kr. - March 20, 2017, 2:02 pm

use function arguments (not global vars) and return values

replace 'sys.stdout.write("")' to 'print "",' and try to use 'if __name__ == "__main__":' to create a clean structure

are u new at programming? Read a book about python and learn about function, classes and everything else about python.

as an example:
learn python the hard way
dive into python

check google about python book pdf. you learn python really fast

obby - March 20, 2017, 11:50 pm

(Mar 15, 2017 11:32 PM)41phafr34k Wrote: [\[CITE\]](#) I wrote a code to parse out words with less than 8 characters and take out duplicates from wordlists. I made few small test files while i wrote it and it worked perfect by the time I finished (I'm no coder... yet... so this small script took me a while).

Anyways the problem I am having now is that I tried to run some larger files through it and it wont even fix the first 10 lines.

Is this something to do with my code or is it due to the file sizes, should I just do it in bash? I wanted to use python because thats what I am trying to teach myself and thought it would be a great project. Anyways, here is the code....

```
Code #!/usr/bin/env python2
```

```
import os
```

```
def parser():
```

```
    f = open(file)
```

```
    output = []
```

```
    for line in f:
```

```
        if len(line) >= 9:
```

```
            output.append(line)
```

```
    f.close()
```

```
    f = open(file, 'w')
```

```
    f.writelines(output)
```

```
    f.close()
```

```
def del_dupes():
```

```
    os.system('sort -u "%s" > "%s"'%(file, new_file))
```

```
print ""Ok, so let's get started...
```

```
This file will parse out all words
```

```
with less then 8 letters and also
```

```
delete duplicates so you may you
```

```
use your wordlists for proper WPA
```

```
cracking. At the end you will have
```

```
a new file ready for WPA cracking.""
```

```
file = raw_input("What is the name of the file to parse?>>> ")
```

```
new_file = 'new_'+file
```

```
parser()
```

```
del_dupes()
```

```
print "We're all done..."
```

```
print "You're new file name is %s" % new_file
```

```
I'm going to try to find out at what point (size) it begins giving issues.
```

I've narrowed down the problem to specific wordlists I made a few of my own by just making 100 lines then copying them over and over and over again then selecting all copy and repeating as necessary to get to 22.7MB which is the smallest size I had trouble with and it worked fine.

But still if anyone see's an issue with how I am going about this, please let me know.

If it's a /huge/ file you'll have to wait for the entire file to load into ram the way you're doing otherwise you can simply iter it and it will use memory as it needs to.

This is done by: for line in open('file.txt', 'r'): instead of storing it in a variable :)