CS 465 PA2

Coversheet

Liye Yang

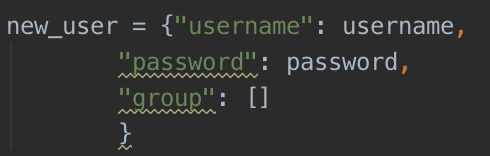
800320031

CODE

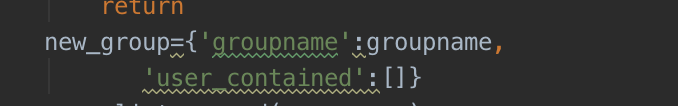
import sys  
  
currentUser = {}  
userlist = []  
grouplist = []  
filelist = []  
  
  
def converTORWX(permi):  
 permission = []  
 i = 0  
 for p in permi:  
 if i == 0 or i == 3 or i ==6:  
 if permi[i] == 0:  
 permission .append('-');  
 else :  
 permission.append("r")  
 elif i== 1 or i==4 or i==7:  
 if permi[i] == 0:  
 permission .append('-');  
 else :  
 permission.append("w")  
 else :  
 if permi[i] == 0:  
 permission .append('-');  
 else :  
 permission.append("x")  
 i = i + 1  
 p1 = ''.join(permission[:3])  
 p2 = ''.join(permission[3:6])  
 p3 = ''.join(permission[6:9])  
  
 return p1+" "+p2+" "+p3  
def isUserInList(username):  
 for user in userlist :  
 if user['username'] == username:  
 return True  
 return False  
def isGrpInList(groupname):  
 for group in grouplist:  
 if group['groupname'] == groupname:  
 return True  
 return False  
def isFileInList(filename):  
 for file in filelist:  
 if file['filename'] == filename:  
 return True  
 return False  
  
def useradd(username, password):  
  
  
 f\_audit = open("audit.txt","a")  
 f = open("account.txt","a")  
  
 if currentUser:  
  
 if currentUser['username'] != "root":  
 print("ERROR:only root user can add group ")  
 f\_audit.write("ERROR:only root user can add group\n")  
 return  
  
  
 if isUserInList(username):  
 print("The username exist, create user failed, please use other name")  
 f\_audit.write("The username exist, create user failed, please use other name\n")  
 return  
 else:  
 new\_user = {"username": username,  
 "password": password,  
 "group": []  
 }  
 userlist.append(new\_user)  
 if new\_user!= None:  
 print("User "+ new\_user['username'] +" created")  
 f\_audit.write("User "+ new\_user['username'] +" created\n")  
 f.write(username+" "+password+"\n")  
 f.close()  
 f\_audit.close()  
  
  
  
def login(username, password):  
 f\_audit = open("audit.txt","a")  
 global currentUser  
  
 up\_list = []  
 for line in open("account.txt"):  
 users, pwd = line.split(" ")  
 user\_pwd = {"name":users,  
 "pwd":pwd}  
 up\_list.append(user\_pwd)  
  
 if currentUser:  
 print("The System can only login one user at a time")  
 f\_audit.write("The System can only login one user at a time\n")  
 return  
 for user in userlist :  
 if user['username'] == username:  
 if user['password'] == password:  
 currentUser = user  
 print("User " + user['username'] +" logged in")  
 f\_audit.write("User " + user['username'] +" logged in\n")  
 return  
 print("ERROR:Password or Username is incorrect ")  
 f\_audit.write("ERROR:Password or Username is incorrect \n")  
 f\_audit.close()  
  
  
  
  
def logout():  
  
 global currentUser  
 f\_audit = open("audit.txt", "a")  
 if currentUser:  
  
 print("User " + currentUser['username'] + " logged out")  
 f\_audit.write("User " + currentUser['username'] + " logged out\n")  
 currentUser= {}  
  
 else:  
 print("ERROR:user must be logged in first!")  
 f\_audit.write("ERROR:user must be logged in first!\n")  
 f\_audit.close()  
  
  
  
def groupadd(groupname):  
 f\_audit = open("audit.txt", "a")  
  
 if currentUser['username'] != "root":  
 print("ERROR:only root user can add group ")  
 f\_audit.write("ERROR:only root user can add group\n")  
 return  
  
  
 if isGrpInList(groupname):  
 print("ERROR:The group name is existed, create group failed, please use other name")  
 f\_audit.write("ERROR:The group name is existed, create group failed, please use other name\n")  
 return  
 if groupname == "nil":  
 print("ERROR:Group can not have the name 'nil'")  
 f\_audit.write("ERROR:Group can not have the name 'nil'\n")  
 return  
 new\_group={'groupname':groupname,  
 'user\_contained':[]}  
 grouplist.append(new\_group)  
 if new\_group!=None:  
 print("Group "+ new\_group['groupname'] +" created")  
 f\_audit.write("Group "+ new\_group['groupname'] +" created\n")  
 f\_audit.close()  
  
  
def usergrp(username,groupname):  
 f\_audit = open("audit.txt", "a")  
 if not isGrpInList(groupname):  
 print("ERROR: The group does not exist")  
 f\_audit.write("ERROR: The group does not exist\n")  
 if not isUserInList(username):  
 print("ERROR: The user does not exist")  
 f\_audit.write("ERROR: The user does not exist\n")  
  
 for user in userlist:  
 if user['username'] == username:  
 user["group"].append(groupname)  
 for group in grouplist:  
 if group['groupname'] == groupname:  
 group["user\_contained"].append(user['username'])  
 print("User "+user['username']+" added to group "+groupname)  
 f\_audit.write("User "+user['username']+" added to group "+groupname+"\n")  
 f\_audit.close()  
  
  
  
def mkfile(filename):  
 f\_audit = open("audit.txt", "a")  
  
 global currentUser  
 if isFileInList(filename):  
 print("ERROR:The file name is existed, create file failed, please use other name")  
 f\_audit.write("ERROR:The file name is existed, create file failed, please use other name\n")  
  
 f = open(filename, "w")  
 new\_file = {"filename":filename,  
 "permissions":[1,1,0,0,0,0,0,0,0],  
 "owner": currentUser,  
 "group":"nil"}  
 filelist.append(new\_file)  
 if new\_file :  
 print("File" + new\_file['filename'] + " with owner " + new\_file['owner']['username'] + "and default "  
 "permissions created")  
 f\_audit.write("File" + new\_file['filename'] + " with owner " + new\_file['owner']['username'] + "and default "  
 "permissions created\n")  
 f\_audit.close()  
  
  
def chmod(filename,rwx1,rwx2,rwx3):  
 f\_audit = open("audit.txt", "a")  
  
  
 permissions =[]  
  
 permissions.append(rwx1)  
 permissions.append(rwx2)  
 permissions.append(rwx3)  
 permission = ''.join(permissions)  
 permi = []  
 for p in permission:  
 if p == 'r' or p == 'w' or p == 'x':  
 permi.append(1)  
 else: permi.append(0)  
  
  
 if not isFileInList(filename):  
 print("ERROR:File does not exist")  
 f\_audit.write("ERROR:File does not exist\n")  
 for file in filelist:  
 if file['filename'] == filename:  
 file['permissions'] = permi  
 print("Permissions for "+file['filename']+"set to "+converTORWX(permi)+" by " +currentUser['username'])  
 f\_audit.write("Permissions for "+file['filename']+"set to "+converTORWX(permi)+" by " +currentUser['username']+"\n")  
 f\_audit.close()  
  
  
def chown(filename,username):  
 f\_audit = open("audit.txt", "a")  
  
 if currentUser['username'] != "root":  
 print("ERROR:only root user can change the owner of a file")  
 f\_audit.write("ERROR:only root user can change the owner of a file\n")  
 return  
 if not isFileInList(filename):  
 print("ERROR:File does not exist")  
 f\_audit.write("ERROR:File does not exist\n")  
 elif not isUserInList(username):  
 print("ERROR:User does not exist")  
 f\_audit.write("ERROR:User does not exist\n")  
 else:  
 for file in filelist:  
 if file['filename'] == filename:  
 preowner = file['owner']  
 for user in userlist:  
 if username == user['username']:  
 file['owner'] = user  
 print( "Owner of "+preowner['username']+" changed to "+file['owner']['username'])  
 f\_audit.write("Owner of "+preowner['username']+" changed to "+file['owner']['username']+"\n")  
  
 f\_audit.close()  
  
  
  
  
def chgrp(filename,groupname):  
 f\_audit = open("audit.txt", "a")  
  
  
 if not isFileInList(filename):  
 print("ERROR:File does not exist")  
 f\_audit.write("ERROR:File does not exist\n")  
 elif not isGrpInList(groupname):  
 print("ERROR:Group does not exist")  
 f\_audit.write("ERROR:Group does not exist\n")  
 else:  
 for file in filelist:  
 if file['filename'] == filename:  
 owner = file['owner']  
 if currentUser['username'] != "root" and currentUser['username'] != owner:  
 print("ERROR current user is not the owner and not root")  
 f\_audit.write("ERROR current user is not the owner and not root\n")  
 return  
 for group in owner['group']:  
 if group == groupname or groupname == "nil" or currentUser == "root":  
 file['group'] = groupname  
 print("Group for "+file['filename']+ "set to "+group+" by "+owner['username'])  
 f\_audit.write("Group for "+file['filename']+ "set to "+group+" by "+owner['username']+"\n")  
 f\_audit.close()  
  
  
  
def read(filename):  
 f\_audit = open("audit.txt", "a")  
  
 f = open(filename,"r")  
 if not isFileInList(filename):  
 print("ERROR:File does not exist")  
 f\_audit.write("ERROR:File does not exist\n")  
 for file in filelist:  
 if file['filename'] == filename:  
 if currentUser['username'] == "root":  
 print("User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read())  
 f\_audit.write("User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read()+"\n")  
 elif currentUser == file['owner'] and file['permissions'][0] == 1:  
 print("User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read())  
 f\_audit.write( "User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read()+ "\n")  
 elif file['group'] in currentUser['group'] and file['permissions'][3] == 1:  
 print("User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read())  
 f\_audit.write(  
 "User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read() + "\n")  
 elif file['permissions'][6] == 1:  
 print("User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read())  
 f\_audit.write(  
 "User " + currentUser['username'] + " read " + file['filename'] + " as:" + f.read() + "\n")  
 else:  
 print("Error, Access Denied")  
 f\_audit.write("Error, Access Denied\n")  
 f.close()  
 f\_audit .close()  
  
  
def write(filename, \*text):  
 f\_audit = open("audit.txt", "a")  
 text = ' '.join(text[0])  
 f = open(filename,"a")  
 if not isFileInList(filename):  
 print("ERROR:File does not exist")  
 f\_audit.write("ERROR:File does not exist\n")  
 for file in filelist:  
 if file['filename'] == filename:  
 if currentUser == "root":  
 f.write(text+"\n")  
 print("User " + currentUser['username']+" wrote to "+file['filename'] + ":" + text)  
 f\_audit.write("User " + currentUser['username']+" wrote to "+file['filename'] + ":" + text+"\n")  
 elif currentUser == file['owner'] and file['permissions'][1] == 1:  
 f.write(text+"\n")  
 print("User " + currentUser['username'] + " wrote to " + file['filename'] + ":" + text)  
 f\_audit.write("User " + currentUser['username'] + " wrote to " + file['filename'] + ":" + text + "\n")  
 elif file['group'] in currentUser['group'] and file['permissions'][4] == 1:  
 f.write(text+"\n")  
 print("User " + currentUser['username'] + " wrote to " + file['filename'] + ":" + text)  
 f\_audit.write("User " + currentUser['username'] + " wrote to " + file['filename'] + ":" + text + "\n")  
 elif file['permissions'][7] == 1:  
 f.write(text+"\n")  
 print("User " + currentUser['username'] + " wrote to " + file['filename'] + ":" + text)  
 f\_audit.write("User " + currentUser['username'] + " wrote to " + file['filename'] + ":" + text + "\n")  
 else:  
 print("Error, Access Denied")  
 f\_audit.write("Error, Access Denied\n")  
 f.close()  
 f\_audit.close()  
  
def execute(filename):  
 f\_audit = open("audit.txt", "a")  
 f = open(filename)  
 if not isFileInList(filename):  
 print("ERROR:File does not exist")  
 f\_audit.write("ERROR:File does not exist\n")  
 return  
 for file in filelist:  
 if file['filename'] == filename:  
 if currentUser == "root":  
 print("FIle "+file['filename']+" execute by"+currentUser['username'])  
 f\_audit.write("FIle "+file['filename']+" execute by"+currentUser['username']+"\n")  
 elif currentUser == file['owner'] and file['permissions'][2] == 1:  
 print("FIle " + file['filename'] + " execute by" + currentUser['username'])  
 f\_audit.write("FIle " + file['filename'] + " execute by" + currentUser['username']+"\n")  
 elif file['group'] in currentUser['group'] and file['permissions'][5] == 1:  
 print("FIle " + file['filename'] + " execute by" + currentUser['username'])  
 f\_audit.write("FIle " + file['filename'] + " execute by" + currentUser['username']+"\n")  
 elif file['permissions'][8] == 1:  
 print("FIle " + file['filename'] + " execute by" + currentUser['username'])  
 f\_audit.write("FIle " + file['filename'] + " execute by" + currentUser['username']+"\n")  
 else:  
 print("ERROR:User " + currentUser['username']+ " denied execute access to" + file['filename'])  
 f\_audit.write("FIle " + file['filename'] + " execute by" + currentUser['username']+"\n")  
 f.close()  
 f\_audit.close()  
  
  
def ls(filename):  
 f\_audit = open("audit.txt", "a")  
 if currentUser:  
 if not isFileInList(filename):  
 print("ERROR:File does not exist")  
 f\_audit.write("ERROR:File does not exist\n")  
 for file in filelist:  
 if filename == file['filename']:  
 print(file['filename']+": "+ file['owner']['username'] +" "+ file['group']+" "+converTORWX(file['permissions']))  
 f\_audit.write(file['filename']+": "+ file['owner']['username'] +" "+ file['group']+" "+converTORWX(file['permissions'])+"\n")  
 else:  
 print("ERROR:user must be logged in first!")  
 f\_audit.write("ERROR:user must be logged in first!\n")  
 f\_audit.close()  
  
def end():  
  
 file\_f = open("file.txt","w")  
 group\_f = open("group","w")  
  
 for group in grouplist:  
 group\_f.write(group['groupname']+": ")  
 for user in group['user\_contained']:  
 group\_f.write(user + " ")  
  
 for file in filelist:  
 file\_f.write(file['filename'] + ": " + file['owner']['username']+" " + file['group']+" ")  
 file\_f.write(converTORWX(file['permissions'])+" \n")  
  
 file\_f.close()  
 group\_f.close()  
  
  
def main(argv):  
 f\_audit = open("audit.txt","w")  
 f\_audit.close()  
 f\_account = open("account.txt","w")  
 f\_account.close()  
 f = open(argv[1], "r")  
 commands = f.read()  
 commands = str(commands).split('\n')  
 f.close()  
  
 for command in commands :  
 command = str(command).split(' ')  
 print(command)  
 commandline(command)  
  
def commandline(\*parameters):  
  
 global currentUser  
 parameters = parameters[0]  
  
 if not userlist:  
 if parameters[0] == "useradd":  
 if parameters[1] == "root":  
 useradd(parameters[1], parameters[2])  
 else:  
 print("ERROR:The first user can only be root")  
 else:  
 print("ERROR:There is no root user,the first common can only be 'useradd root password'to add a root user")  
  
 else:  
 if parameters[0] == "useradd":  
 useradd(parameters[1], parameters[2])  
  
 if parameters[0] == "login":  
 login(parameters[1], parameters[2])  
  
 if parameters[0] == "logout":  
 logout()  
  
 if parameters[0] == "groupadd":  
 groupadd(parameters[1])  
  
 if parameters[0] == "usergrp":  
 usergrp(parameters[1], parameters[2])  
  
 if parameters[0] == "mkfile":  
 mkfile(parameters[1])  
  
  
 if parameters[0] == "chmod":  
 chmod(parameters[1], parameters[2], parameters[3], parameters[4])  
  
 if parameters[0] == "chown":  
 chown(parameters[1], parameters[2])  
  
 if parameters[0] == "chgrp":  
 chgrp(parameters[1], parameters[2])  
  
 if parameters[0] == "read":  
 read(parameters[1])  
  
 elif parameters[0] == "write":  
 write(parameters[1], parameters[2:])  
  
 if parameters[0] == "execute":  
 execute(parameters[1])  
  
 if parameters[0] == "ls":  
 ls(parameters[1])  
  
 if parameters[0] == "end":  
 end()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main(sys.argv)

Explanation:

The format of “accounts.txt” is “username password” I used a python function split(“ ”)to split username and password. Finally assign the value to this structure.



The data structure of group



The data structure of file.



I divide permission into a int type list, each element stand for ‘r’ or ‘w’ or ‘x’

And I use converTORWX() function convert int number list to a rwx string list.



Audit.txt of test case 1

User root created  
User root logged in  
User alice created  
User bob created  
Group students created  
User bob added to group students  
User bob added to group students  
User tom created  
User root logged out  
User alice logged in  
Filefile1.txt with owner aliceand default permissions created  
User alice wrote to file1.txt:Text from Alice in file1  
Filefile2.txt with owner aliceand default permissions created  
User alice wrote to file2.txt:Text from Alice in file2  
Permissions for file2.txtset to rw- rw- r-- by alice  
ERROR current user is not the owner and not root  
User alice logged out  
User tom logged in  
Error, Access Denied  
User tom read file2.txt as:  
Error, Access Denied  
The System can only login one user at a time  
User tom logged out  
ERROR:Password or Username is incorrect   
User bob logged in  
Error, Access Denied  
User bob read file2.txt as:  
User bob logged out  
User root logged in  
Owner of alice changed to tom  
file1.txt: tom nil rw- --- ---  
User root logged out  
User tom logged in  
User tom wrote to file1.txt:Text from Tom in file1  
User tom read file1.txt as:  
User tom logged out

Account.txt of test case 1

root ya84\*\_o  
alice Wvu\_4\_Life  
bob SHHHsecret  
tom geek\_247

file .txt of test case 1

file1.txt: tom nil rw- --- ---   
file2.txt: alice nil rw- rw- r--

file1.txt

Text from Alice in file1  
Text from Tom in file1

File2.txt

Text from Alice in file2

group.txt of test case 1

students: alice bob

Au dit.txt of test case 2

User root created  
User root logged in  
User steveo created  
User rita created  
Group testers created  
The username exist, create user failed, please use other name  
User root logged out  
ERROR:Password or Username is incorrect   
User steveo logged in  
Filescript with owner steveoand default permissions created  
User steveo wrote to script:echo Hello World!  
Permissions for scriptset to -wx rw- --x by steveo  
ERROR:File does not exist  
ERROR current user is not the owner and not root  
FIle script execute bysteveo  
User steveo logged out  
ERROR:user must be logged in first!  
User root logged in  
The username exist, create user failed, please use other name  
ERROR:The group name is existed, create group failed, please use other name  
User rita added to group testers  
User rita added to group testers  
ERROR:File does not exist  
FIle script execute byroot  
User root logged out  
User rita logged in  
ERROR:The file name is existed, create file failed, please use other name  
Filescript with owner ritaand default permissions created  
Error, Access Denied  
User rita wrote to script:echo Hello World!  
Error, Access Denied  
User rita read script as:  
FIle script execute byrita  
FIle script execute byrita  
ERROR:only root user can add group  
ERROR:only root user can add group  
User rita logged out

Accout.txt of case 2

root @r00t(705)  
steveo d@man^304  
rita #101'holla!

Group.txt of case 2

testers: rita steveo

file.txt

script: steveo nil -wx rw- --x   
script: rita nil rw- --- ---

script.txt

echo Hello World!