This is my final project submission for NTS370. This project will not be posted on Github and is a current project I am working on for the school as a student ambassador to automate my tasks.

To not impede on an non disclosure documents I have signed and for privacy of upcoming students information no personal student information will be used for demonstration. This means I will only be using example files provided by Encoura and will not be able to provide a demonstration for the script that handles our pdf funnel. These scripts and the idea was made by me but as a disclaimer I did receive help for the python script early on by Jordan Brown. These scripts have taken some time and modeling to UAT’s ACT and SAT process and were made from scratch.

First comes the master script as it is the first thing that is ran to perform these functions.

#!/bin/bash

#runs python script first to create xls for the act bash script to have needed contents.

/usr/bin/python3.6 /ACTAuto.py

#Runs bash script after python script performing the pdf extraction and naming process for students.

./ACTauto.bash

Here is the initial python script “ACTAuto.py” that is ran to extract information from the template .txt file that I will be using for this project demonstration.

#!/usr/bin/python3.6

import xlrd

import xlwt

import os

from datetime import datetime

#get todays date for import naming convention

d = datetime.date(datetime.now())#input("Today's Date MM/DD/YYYY: ")

ddaystring = ""

dmonthstring = ""

dday = d.day

if dday < 10:

ddaystring = "0" + str(dday)

else:

ddaystring = str(dday)

dmonth = d.month

if dmonth < 10:

dmonthstring = "0" + str(dmonth)

else:

dmonthstring = str(dmonth)

dyearstring = str(d.year-2000)

#keep count of errors

errors = 0

#make new workbook

newwb = xlwt.Workbook()

newsheet = newwb.add\_sheet('Sheet1')

#make default font

font = xlwt.Font()

font.name = "Calibri"

font.height = 11\*20

style = xlwt.XFStyle()

style.font = font

#write the headers to the sheet

newsheet.write(0, 0, "First Name", style)

newsheet.write(0, 1, "Last Name", style)

newsheet.write(0, 2, "High School Graduation Year", style)

newsheet.write(0, 3, "Phone Number", style)

newsheet.write(0, 4, "Street Address", style)

newsheet.write(0, 5, "City", style)

newsheet.write(0, 6, "State", style)

newsheet.write(0, 7, "Postal Code", style)

newsheet.write(0, 8, "Email", style)

newsheet.write(0, 9, "CEEB Code", style)

newsheet.write(0, 10, "ACT Score", style)

newsheet.write(0, 11, "Lead Source", style)

#make the date format

date\_format = xlwt.XFStyle()

date\_format.num\_format\_str = 'mm/dd/yyyy'

date\_format.font = font

#file1 = open("nameholder2.txt", "w")

#i will be iterator

i = 0

#open the target txt file

with open("target.txt", "r") as ifile:

#loop through the lines

for line in ifile:

i += 1

#write firstname

s = line[27:43]

s = s.rstrip()

newsheet.write(i, 0, s.title(), style)

#adds this specific portion to text file for naming convention

file1 = open("nameholder2.txt", "a")

print(s, file=file1)

#write lastname

s = line[2:27]

s = s.rstrip()

newsheet.write(i, 1, s.title(), style)

#adds this specific portion to text file for naming convention

file1 = open("nameholder2.txt", "a")

print(s, file=file1)

#write gradyear

s = line[222:226]

newsheet.write(i, 2, s, style)

#write phone

s = line[106:116]

newsheet.write(i, 3, s, style)

#street

s = line[44:84]

s = s.rstrip()

newsheet.write(i, 4, s.title(), style)

#city

s = line[116:141]

s = s.rstrip()

newsheet.write(i, 5, s.title(), style)

#state

s = line[143:145]

newsheet.write(i, 6, s, style)

#zip

s = line[145:150]

newsheet.write(i, 7, s, style)

#email

s = line[550:604]

s = s.rstrip()

newsheet.write(i, 8, s.title(), style)

#hs

s = line[204:210]

newsheet.write(i, 9, s, style)

#score

s = line[268:270]

newsheet.write(i, 10, s, style)

newsheet.write(i, 11, "ACT", style)

#saves the act information as an xl file

newwb.save("ACT Import - " + dmonthstring + "." + ddaystring + "." + dyearstring + ".xls")

print("Successfully ran with " + str(errors) + " errors!")

#os.system("pause")

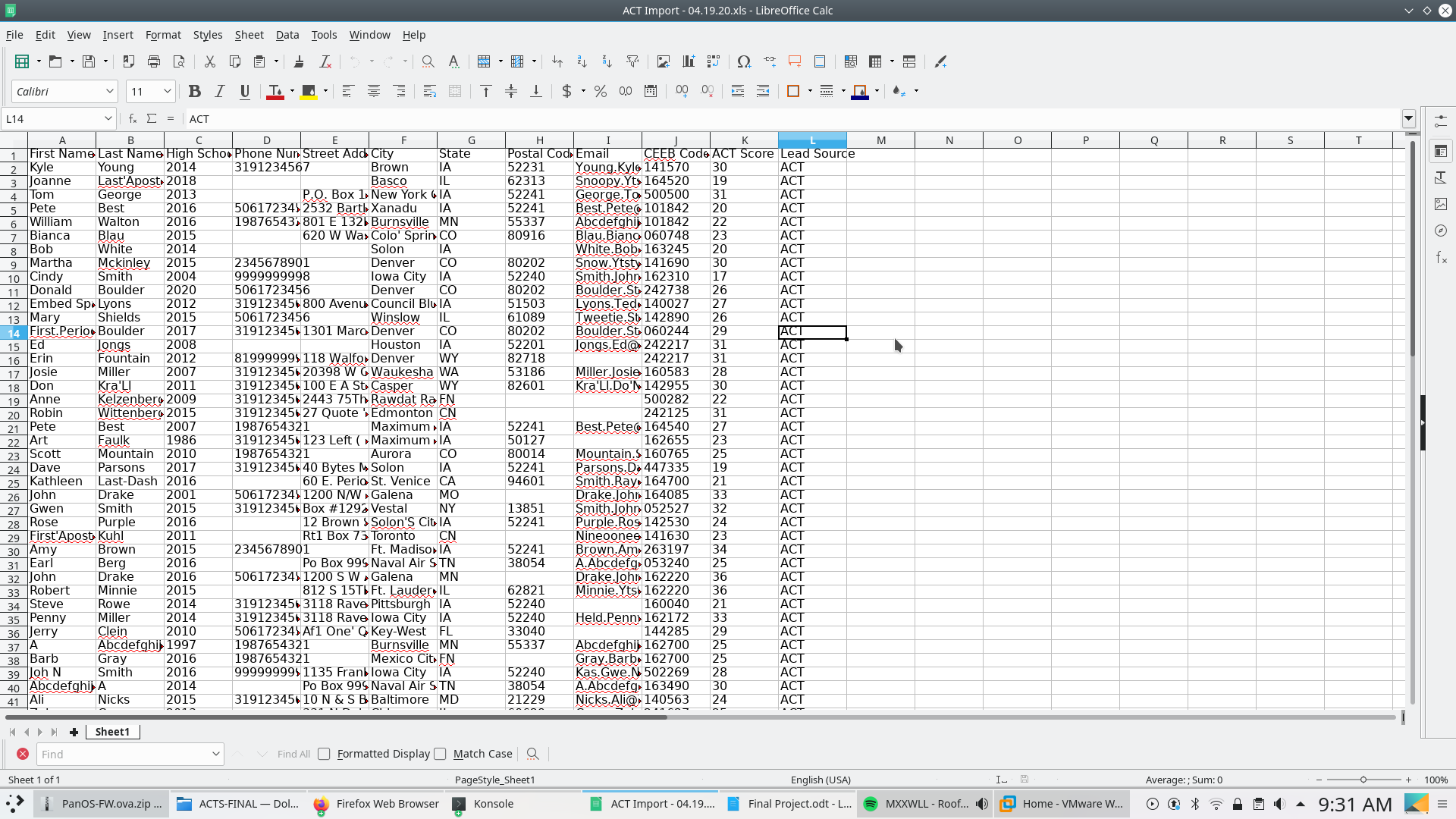
target.txt is the file that is given by Encoura with a text file and a very odd format with all the information of students. This needs to be parsed otherwise it is not use to the school. Here is an example of the student information including two test students (Again disclaimer these are fake individuals with fake grades etc.

19YOUNG KYLE F US6F12 10225973191234567BROWN 16IA52231203019970225 14157040304040 37520142015090915 2930303030 119 -- -- -- -- -- -------- 2 135615141516161615 098097087083087099099093085------------ 2013222093 YOUNG.KYLE@YAHOO.COM 087---------------------------97------------------31------------------0910970910970099097 3609912121212 32099 30095 24 087086088098099092085090095089095095

19LAST'APOSTROPHE JOANNE US6F08-914000162072901 BASCO 14IL62313 20010729 164520 ---20182015101015 2117231419 075 -- -- -- -- -- -------- S 135610110909091212 013062031035019017018052056------------ 03081816421 1 1 SNOOPY.YTSTYPE@YAHOO.COM 1 3 3 ----------------------------------------------------------------------03101105501 0056011 2006805060607 21066 16020 12 056061042037040067066058037066013043

19GEORGE TOM EP.O. BOX 123 US4M11 1 NEW YORK CITY 16IA52241 3010 34 50050038354040 38120132013050513 S 3024363231 122 59 62 62 60 49 6199 3 135615161212131818 087098053064076099099818989834587 26302520 1 1 252252 115 1 1 113534 212115024 84643251714 1735555 8 11 1 1 1 1 1 1111 1 GEORGE.TOM@YAHOO.COM 12521 111211113213111211312313313313344 4334 4 444 44 4 4 4 4 ----------------------------------------------------------------------0930630990990 098 ----- 095099087097067072082099099092073099097096

Now after extracting this data it is funneled into the excel file which looks like this.



After this information is formed there is also a portion that is funneled into a text file called nameholder2.txt with each first name and last name stored on a new line. Below included is the bash script that I wrote to create each individual pdf based on the names provided by the python script. These scripts have been fully tested and are functional and I use them on a daily basis to finish my tasks.

#!/bin/bash

#these are optional commented out options for user input

#echo "what is the path to the file?"

#read file

#grabs file named target2 pdf and assigns it to the variable file.

file="/home/pampew10/Desktop/ACTS/target2.pdf"

#grabs dates to open import previously created by the python script.

DD=$(date +%d)

MM=$(date +%m)

YY=$(date +%y)

echo "Today is Day:$DD Month:$MM Year:$YY"

xdg-open "ACT Import - $MM.$DD.$YY.xls"

#variables for use throughout for loop.

end=2

start=1

start2=2

start3=1

AWK=1

#for loop to funnel through each act individual to make a pdf.

for start3 in {1..100}

do

echo "$start-$end"

#funnels student names provided by python script in nameholder2.txt to nameholder.

#The reason this is needed is each student has a file format that involves the name.

awk 'NR%2{printf "%s ",$0;next;}1' nameholder2.txt | sed -n "$AWK"p | cat > nameholder

#Assigns ACTname varible to the cat output of nameholder.

ACTname=$(cat nameholder)

#cleans up names and changes format from spaces to \_

echo "$ACTname" | sed -e 's/\_/ /g' | awk '{ t=$1; $1=$NF; $NF=t; print}' | sed -e 's/ /\_/g' | cat > nameholder

#assigns new variable to nameholder cat output

ACTname2=$(cat nameholder)

#overwrites nameholder content with new content preceeded by ACT for file format.

echo "ACT $ACTname2" | cat > nameholder

#cat nameholder should output "ACT testname\_testname"

cat nameholder

#Assigns new and final variable to cat output.

ACTname3=$(cat nameholder)

#precontext: All student pdfs are in a giant pdf file with each student having two pages. 1-2 3-4 5-6 etc.

#grabs pages needed for the individuals pdf student file based on start and end variable which add two with each rotation. Saves the new student files based on all paremeters from master file to output location.

pdftk $file cat $start-$end output /home/pampew10/Desktop/ACTS/ACTS2/"$ACTname3".pdf

#adds all the needed variables for loop to proceed.

start3=$(($start3+2))

start2=$(($start2+2))

start=$(($start+2))

end=$(($end+2))

AWK=$(($AWK+1))

done

exit

Now this ends the ACT scripts I have also created bash scripts to perform similar tasks but for SAT. These SAT scripts are not completely functional and are based on a scanned in physical file. Thus I had to use OCR to gain the information from it so it is not 100% accurate.

Here is the first script which performs the ocr and pdf2text process. These have been wonderful in aiding this process and pdf2text does an awesome job at performing the task it does in an easier to use format. Now these SAT scripts could have been done in Python but I thought it’d be fun at the time to do it in bash and will probably rewrite in Python at some point.

#!/bin/bash

#lists the directory and only includes the pdfs.

ls | grep ".pdf" > test2.txt

AWK=1

start3=1

#for loop for funneling through the pdfs and performing tasks.

for start3 in {1..100}

do

#assigns variable test3 to the catted test2.txt previously created by the list of the directories pdfs with a line by line format assigned by sed.

test3=$(cat test2.txt | sed -n "$AWK"p)

#echos previously made variable.

echo $test3

#runs individual file through ocrmypdf and resaves as same file.

ocrmypdf "$test3" "$test3"

#assigns new variable to file but instead of pdf to txt.

test4=$(echo "$test3" | sed 's/.pdf/.txt/g')

#cats new variable aka txt file and funnels into a master file.

cat "$test4" >> master.txt

#runs newly saved copiable pdf from ocr through pdf2text and saves as new text file.

pdf2txt -o "$test4" "$test3"

start3=$((start3+1))

AWK=$(($AWK+1))

done

Just for reference this is the xls converter for the information from the new text files from the ocr etc. This is a bash script I wrote to perform the extraction process of the information I could get from the text files based on the pdf images. This is only about halfway done and is not properly commented yet but I thought I would include.

#!/bin/bash

ls | grep ".pdf" > test2.txt

echo "First Name, Last Name, High School Graduation Year, Phone Number, Street Address, City, State, Postal Code, Email, Company, SAT Score- 1600, Lead Source" >> importtest.csv

AWK=1

start3=1

for start3 in {1..100}

do

#grabs each line from text file individually for every loop.

test3=$(cat test2.txt | sed -n "$AWK"p)

#echos output for test3

echo $test3

#Transition to pdf to txt files being obtained.

test4=$(echo "$test3" | sed 's/.pdf/.txt/g')

#funnels txt files output to a master txt file.

cat "$test4" >> master.txt

email=$(cat "$test4" | grep -B 1 "Email Address" | sed 's/Email Address//g' | sed '/^$/d' | awk '{for(i=1;i<=NF;i++) {if (length($i)>3) { printf "%s ", $i }} }')

#company=$(cat "$test4" | grep -B 1 "School Name" | sed 's/School Name//g')

company=$(cat "$test4" | grep -A 1 "High School Information" | sed 's/High School Information//g' | sed '/^$/d')

phonenumber=$(cat "$test4" | grep -B 1 "Phone Number" | sed -ne '/--/,$ p' | sed 's/Phone Number//g' | tr -dc '0-9, -' | sed 's/^..//' | awk '{for(i=1;i<=NF;i++) {if (length($i)>3) { printf "%s ", $i }} }' | sed '/^$/d')

lastname=$(cat "$test4" | grep -B 1 "Last Name" | sed 's/Last Name//g' | sed '/^$/d')

score=$(cat "$test4" | grep -A 2 "Total" | sed 's/Total//g' | sed '/^$/d' | sed -ne '/--/,$ p' | sed '1,/--/!d' | sed 's/--//g' | sed '/^$/d' | awk '{for(i=1;i<=NF;i++) {if (length($i)<4) { printf "%s ", $i }} }' | tr -dc '0-9' | awk '{for(i=1;i<=NF;i++) {if (length($i)>2) { printf "%s ", $i }} }')

address=$(cat "$test4" | grep -A 4 -e "Student Information" -e "StudentInformation" | sed 's/Student Information//g' | sed 's/StudentInformation//g' | sed -n 2p)

address2=$(cat "$test4" | grep -A 4 -e "Student Information" -e "StudentInformation" | sed 's/Student Information//g' | sed 's/StudentInformation//g' | sed -n 4p)

#email=$(pdfgrep -n "@" "$test3" | grep -o -P '.{0,50}@.{0,50}' | tr -d ' ' | sed -n 2p)

#name=$(pdfgrep -n "University of Advancing Technology" "$test3" | sed 's/University of Advancing #Technology//g' | awk '{for(i=1;i<=NF;i++) {if (length($i)>1) { printf "%s ", $i }} }')

#email=$(pdfgrep -n "@" "$test3" | grep -o -P '.{0,50}@.{0,50}' | tr -d ' ' | sed -r '/^.{,4}$/d')

#FINISHED - grabs the score of the student

#score=$(pdfgrep -n "Total" "$test3" | sed -n 2p | sed 's/Total//g' | tr -d ' ' | sed 's/2://g' | tr -dc '0-9' | awk '{for(i=1;i<=NF;i++) {if (length($i)<4) { printf "%s ", $i }} }')

#echo $email

#echo $name

echo "'First Name', $lastname, 'High School Graduation Year', $phonenumber, $address, $address2, 'State', 'Postal Code', $email, $company, $score, SAT" >> importtest.csv

start3=$((start3+1))

AWK=$(($AWK+1))

done

pandoc master.txt -o master.xls