Installation and Getting Started

-Note: This section abstracted from the main paper to be included in software file.

Getting started with the application is simple and only requires a couple steps. You will need to have three things installed on your system to successfully run the program. The first is you must have Python 2.7 installed. The second requirement is to install two python components. You will need python mysql connector and python mysqldb. If you happen to be using a Debian based Linux distro you can at the command line from root use the commands "*apt-get install python-mysql.connector*" and "*apt-get install python-mysqldb*". If you are using red-hat or another distro you may have to look up the instructions. Finally you will also need to already have procmail installed on your smtp server.

The CLI

The CLI is developed to run on a Linux machine running python. The project description indicates it is to run on a Linux Operating system and the CLI has been tested to run on flip which is RedHat, in an Ubuntu, and in CentOs 7, and Mac OS X 10.9.5 operating systems which is actually Unix based not Linux. We believe it to also run in other Mac OS versions but this remains untested. The interface runs a Python based Curses implementation that interfaces with the back end database to display appointments for the particular user.

The application name is cli.py and the file has been made executable so all you need do at the Linux command line is type ./cli.py.

After you launch the application for the first time, you'll be met with an opening screen. Enter 'M' to go to the menu and enter '1' to register a new account.

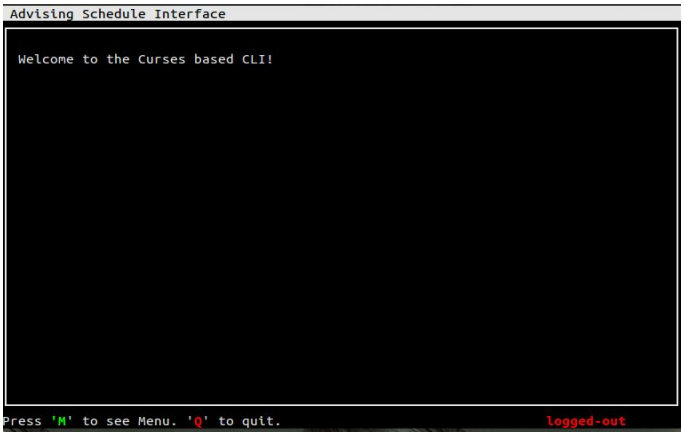


Fig 6. Opening Screen

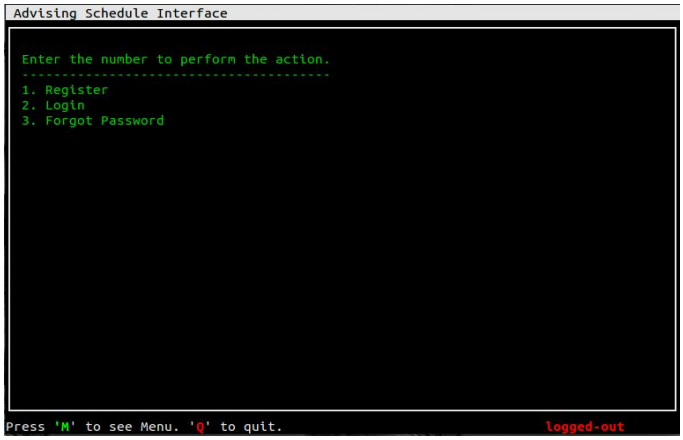


Fig 7. Initial Menu

You MUST register with the EECS email that receives the advising appointment requests. The filter pulls this email address from the original email sent by the advising system, and stores the appointment in the database with this email as a key. Therefore, in order to view your advising appointments, you must use this email address in your registration. See figure 9 below for a sample email sent by the advising system (use the email in the "To" line).

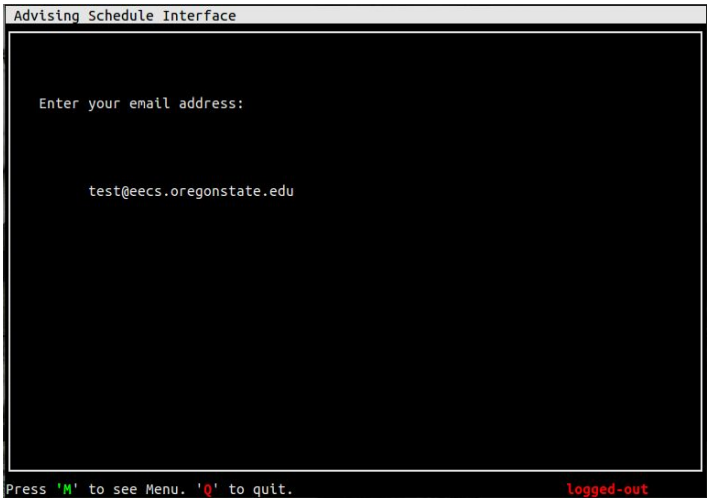


Fig 8. Enter your Registration email

From: do.not.reply@engr.orst.edu
Sent: Monday, November 26, 2012 19:15
To: REDACTED@oregonstate.edu; dmcgrath@eecs.oregonstate.edu
Subject: Advising Signup with McGrath, D Kevin confirmed for REDACTED

Advising Signup with McGrath, D Kevin confirmed
Name: REDACTED
Email: REDACTED@oregonstate.edu
Date: Wednesday, November 21st, 2012
Time: 1:00pm - 1:15pm

Please contact support@engr.oregonstate.edu if you experience problems

Fig 9. Sample email sent by advising system

The application will then send you a temporary password to that email, which you can use to sign in to the application. After you receive the confirmation screen in Figure 10 below, check your email and copy the temporary password from the email.

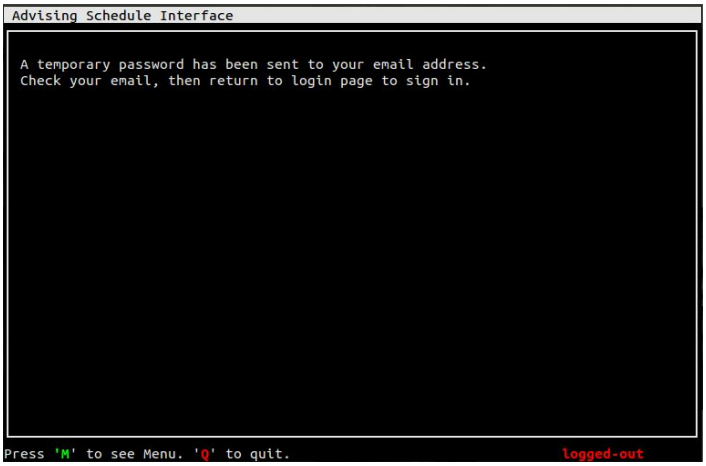


Fig 10. Temporary Password notification

Return back to the CLI and go to the login screen by entering 'M' followed by '2'. After you sign in for the first time, you will be prompted to change your password with one of your choosing. You'll be required to enter your new password twice, and upon successful completion, will be notified with the message in Figure 11. After you receive this confirmation, enter 'M' to be redirected to the user's main menu.

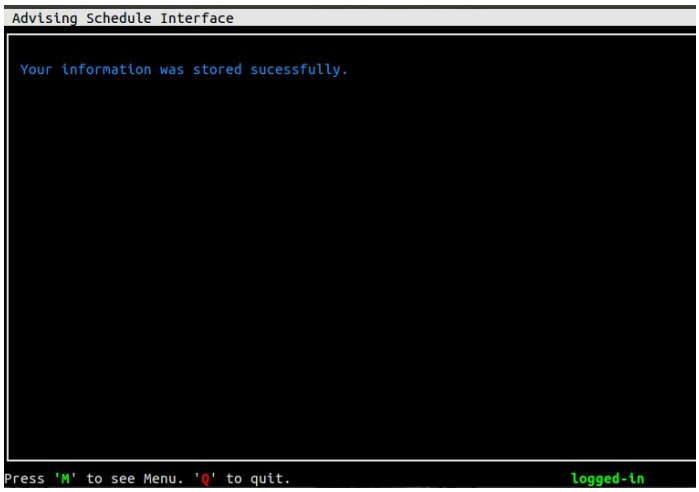


Fig 11. Change Password Confirmation

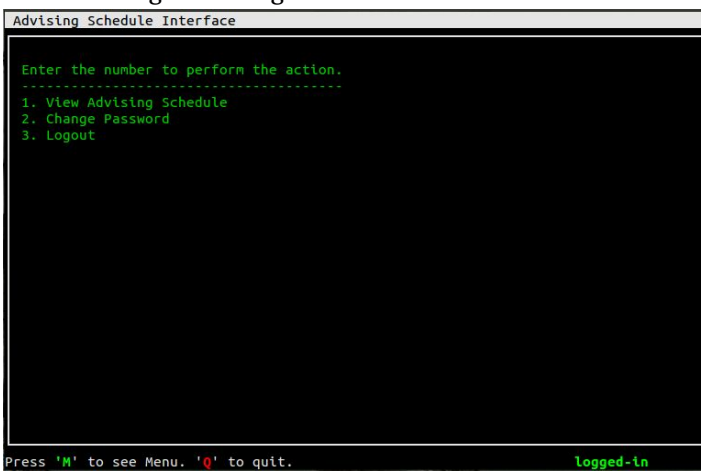


Fig 12. User's Main Menu

Now that you are registered with the CLI application and have implemented your .procmailrc script, you are ready to start building your database of advising appointments. The CLI will only pull forward appointments that have been generated through the procmail script, so any existing appointments you may have had prior to installing this application will not appear in the CLI. Below we'll demonstrate how to use the CLI to view your advising appointments and request appointment cancellations.

Figure 13 shows the Appointments Summary full of dummy student information. This is the view when you choose menu item 1 in figure 12 above. Here you can use the up and down arrow on your keyboard to cycle through the appointments. The appointments are displayed in descending date order. This set up should allow you to cycle through appointments quickly and efficiently. Figure 14 shows the screen you'll see when you select a particular appointment. We pull this up by simply choosing the reference number in the Appointments Summary screen. From here you can cancel the appointment or drop back out. If you choose to cancel the appointment follow the prompts. An email is then sent to both you and the student notifying you of the cancellation.

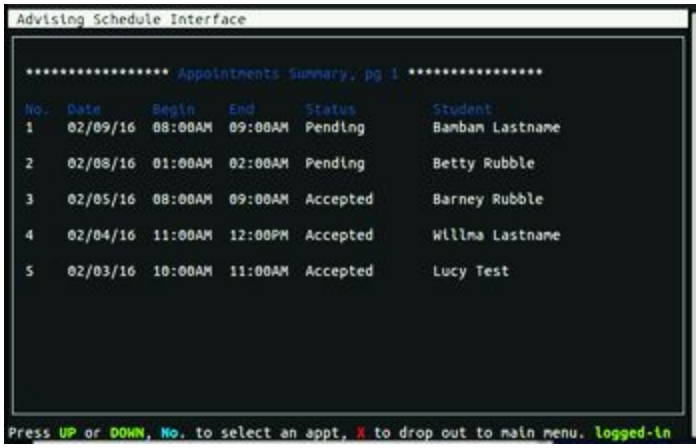


Fig 13. User's Appointments Summary page,

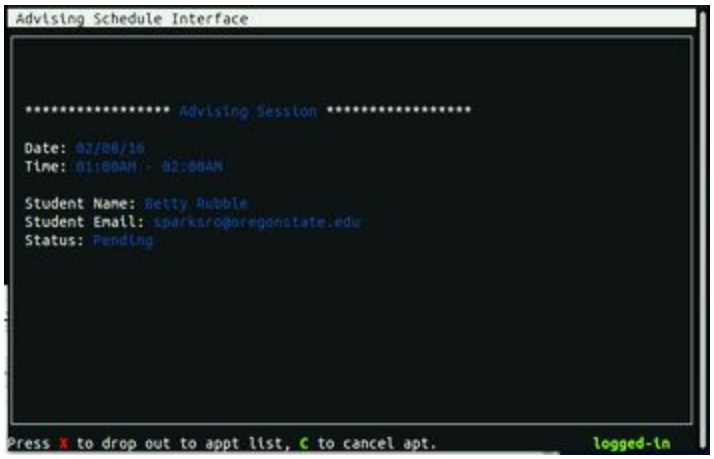


Fig 14. Individual Advising Session

3.2 The Production Procmail Filter

The first step of setting up the procmail filter script is to add the advisingProcmailFilter.py script to your home directory. Next you will need to add the following rules to your .procmailrc file:

```
:0fwHB:
* ^FROM.*<your email address>
| /usr/bin/python $HOME/advisingProcmailFilter.py

:0fwHB:
* ^FROM.*do.not.reply@engr.orst.edu
| /usr/bin/python $HOME/advisingProcmailFilter.py

:0fwHB:
* ^FROM.*procmailtestscs@gmail.com
| /usr/bin/python $HOME/advisingProcmailFilter.py

:0:
* ^FROM.*
```

<your desired inbox>

:0
* < 10
/dev/null

Insert your email address in the spot of <your email address> without the < or >. It is important to also follow the ordering of the above recipes. You need the check for your email address to be first before any other recipe as the other advisingProcmailFilter.py script will change the from address to your email address. This is also important to note in case you have other recipes that may conflict with this. The ^FROM.* will move all emails into your desired inbox. You simply need to input the name of your desired inbox in the <your desired inbox> space without the < or >.

The script is now in place to do the work for you. You will now receive the emails from the advising system as you were previously, but they will now have meeting requests attached. Figure 15 is an example of the new sign up email.

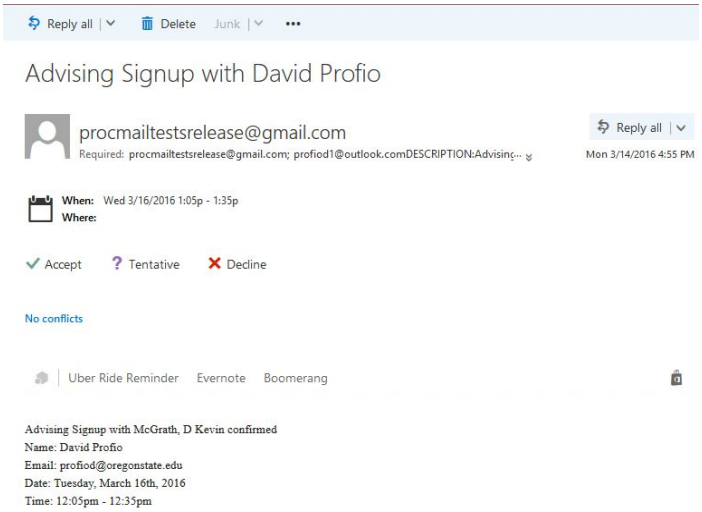


Fig 15. A new appointment notification.

If you would like to accept this appointment, and have it added to your calendar, you just need to hit the green check mark for accept. If you would like remove this request you will simply just need to click decline. Until you address the email, the appointment will show as “Pending” in the CLI. Once you accept it the status will show as “Accepted”. If you choose to decline it the appointment will be removed and no longer show in the CLI. Figure 16 is an example of the new appointment cancellation email.

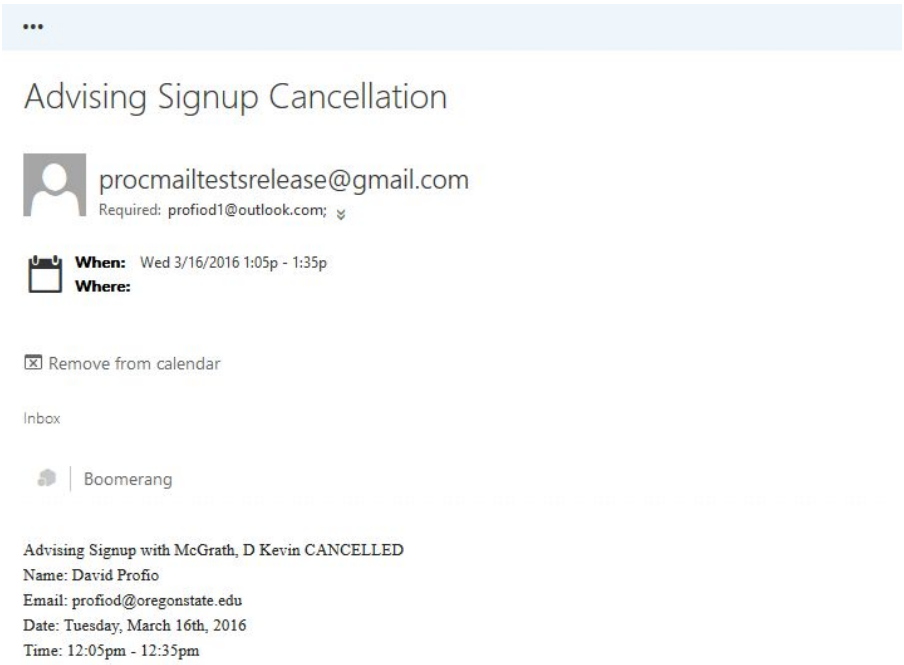


Fig 16. A new cancellation email

The appointment will remain in your calendar until you click the Remove from calendar. Once you click Remove from calendar it will remove the appointment so you no longer see it. It has already been removed from the CLI at this point.

The Testing Procmail Filter

We are including the instructions for the procmail filter that we used for testing in case you would like to simulate how we tested or you do not have access to an outlook inbox that lies behind a linux server with procmail installed. This is important as our ics file is not recognized by Gmail, but is recognized by Outlook as well as Yahoo. This means that Gmail cannot be used to test the validity of our filter's newly created VEVENT.

To begin we will go over connecting to our VM in putty. The host of the VM is:

ubuntu@ec2-52-10-233-116.us-west-2.compute.amazonaws.com

The port is 22. On the left sidebar of putty please expand SSH. You will then need to click on Auth. Please make sure to click on it and not expand it. Go to the private key file for authentication and insert the puttykey2.ppk file that we have included in our submission. The password for this file is "gobeavers!" without quotation marks.

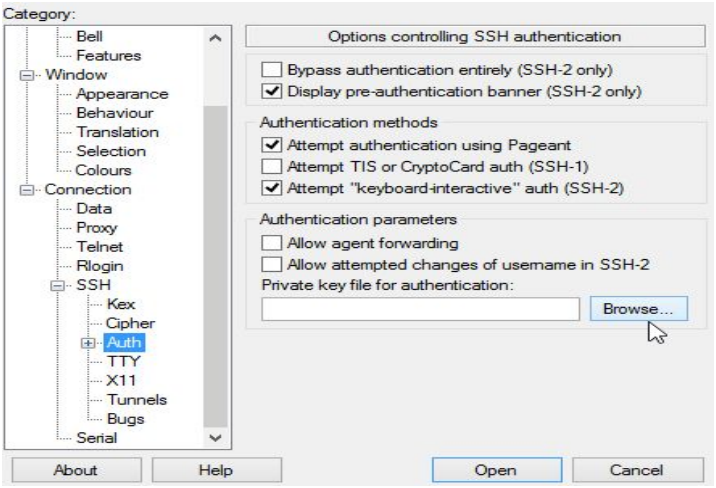


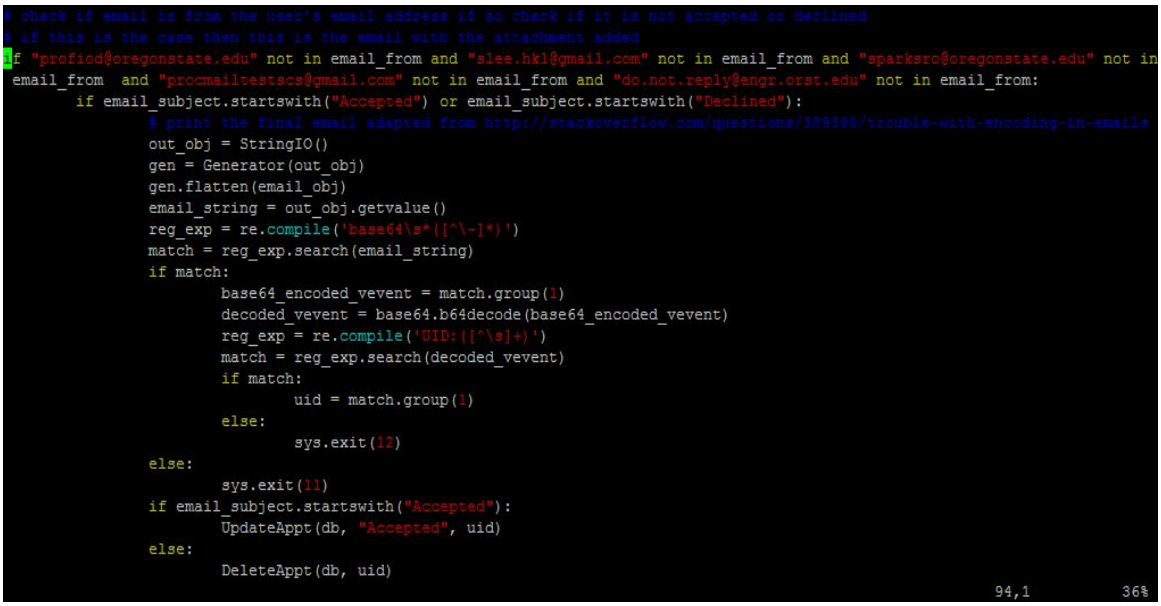
Fig 17. Putty settings

Once you are connected to the VM you should be in the home directory for the user ubuntu. This is where we have housed our .procmailrc file as well as our Mail directory and the testScript.py. Since we do not have access to do.not.reply@engr.orst.edu you will have to use your personal email to act as the Advising System. You will have to add your email in two places. The first is in the .procmailrc file, where we have added a placeholder recipe. Please just change the placeholder email placeholder@placeholder.com to your email address by opening up .procmailrc in vim.

```
:0fwHB:
* ^FROM.*procmailtestsrelease@gmail.com
| /usr/bin/python $HOME/testScript.py

:0fwHB:
* ^FROM.*placeholder@placeholder.com
| /usr/bin/python $HOME/testScript.py

:0fwHB:
* ^FROM.*sparksro@oregonstate.edu
| /usr/bin/python $HOME/testScript.py

:0fwHB:
* ^FROM.*slee.hkl@gmail.com
| /usr/bin/python $HOME/testScript.py

:0fwHB:
* ^FROM.*profiod@oregonstate.edu
| /usr/bin/python $HOME/testScript.py
```

Fig 18. The .procmailrc file on the server

Secondly, you'll have to add you email to the testScript.py. You will need to add the email you are using in location shown in figure 19 below. It is on code line 94 in the testScript.py. To add the check for your email address you will just need to append ‘ and “your_email_address” not in email_from’ without the ‘s right before the colon.



```
# check if email is from the user's email address if so check if it is not accepted or declined
# if this is the case then this is the email with the attachment added
if "profiod@oregonstate.edu" not in email_from and "slee.hkl@gmail.com" not in email_from and "sparksro@oregonstate.edu" not in
 email_from  and "procmailtestscs@gmail.com" not in email_from and "do.not.reply@engr.orst.edu" not in email_from:
        if email_subject.startswith("Accepted") or email_subject.startswith("Declined"):
                # print the final email adapted from http://stackoverflow.com/questions/389398/trouble-with-encoding-in-email
                out_obj = StringIO()
                gen = Generator(out_obj)
                gen.flatten(email_obj)
                email_string = out_obj.getvalue()
                reg_exp = re.compile('base64\s*([^\-]*)')
                match = reg_exp.search(email_string)
                if match:
                        base64_encoded_vevent = match.group(1)
                        decoded_vevent = base64.b64decode(base64_encoded_vevent)
                        reg_exp = re.compile('UID:([^\s]+)')
                        match = reg_exp.search(decoded_vevent)
                        if match:
                                uid = match.group(1)
                        else:
                                sys.exit(12)
                else:
                        sys.exit(11)
                if email_subject.startswith("Accepted"):
                        UpdateAppt(db, "Accepted", uid)
                else:
                        DeleteAppt(db, uid)
                                                                                                94,1          36%
```

Fig 19. The location to add your email address in testScript.py

In order to get the emails into our server you will need to send them to procmailtestsrelease@gmail.com from the email that you entered above (see included sample Request and Cancellation emails). The password for this account is “gobeavers!” without quotations, but you should have no need to go into this account. Once you have sent your email to this account you can run the command “fetchmail” without quotations. This will bring the email from Gmail into our server for procmail to process. If you would like to make sure that the correct email is being printed to standard out in procmail we store all of the emails in Mail/testemails. Please feel free to open this file in vim to verify that only one copy is showing of the advising email.

```
ubuntu@ip-172-31-25-148:~$ fetchmail
1 message for procmailtestsrelease@gmail.com at pop.gmail.com (4116 octets).
procmail: [13033] Tue Mar 15 00:03:08 2016
procmail: Assigning "MAILDIR=/home/ubuntu/Mail"
procmail: Assigning "LOGFILE=/home/ubuntu/.procmaillog"
procmail: Opening "/home/ubuntu/.procmaillog"
reading message procmailtestsrelease@gmail.com@gmail-pop.l.google.com:1 of 1 (41
16 octets) flushed
```

Fig 20. The fetchmail command

```
ubuntu@ip-172-31-25-148:~$ vim Mail/testemails

Content-Type: multipart/mixed; boundary="===============9155953111261411606=="
MIME-Version: 1.0
reply-to: procmailtestsrelease@gmail.com
date: Mon, 14 Mar 2016 20:04:18 -0400
to: profiod1@outlook.com
from: procmailtestsrelease@gmail.com
subject: Advising Signup Cancellation

--===============9155953111261411606==
Content-Type: multipart/alternative;
 boundary="===============4130591056796361731=="
MIME-Version: 1.0

--===============4130591056796361731==
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

Advising Signup with McGrath, D Kevin CANCELLED
Name: David Profio
Email: profiod@oregonstate.edu
Date: Tuesday, March 16th, 2016
Time: 12:05pm - 12:35pm

--===============4130591056796361731==
MIME-Version: 1.0
Content-Type: text/calendar; method="CANCEL"; charset="us-ascii"
Content-Transfer-Encoding: 7bit
```

Fig 21. An example of the testemails file

Another difference between our normal production script and the test script is that we have included code to send a copy of the new email to an outside email address. In our case it is set to profiod1@outlook.com. The password for this account is "gobeavers!" without quotation marks. You can login to it by visiting outlook.com. We use this email address as our test email address since our VEVENTs do not show in Gmail. We set the reply-to for the email to procmailtestsrelease@gmail.com so once you confirm or decline an event in Outlook it will send it back to our testing Gmail account so you can retrieve it using fetchmail and process it.

In our test script, profiod1@outlook.com acts as the advisor's email. You can use this email address when you register with the CLI to view the appointments you have added. However, it is important to note that the CLI is configured to work on a production environment, and when used in conjunction with the the test script won't behave as expected in some situations. One situation is when cancelling an appointment through the CLI. An email will be sent to the student and to profiod1@outlook.com indicating the cancellation, however this email will not be caught by the filter and then converted to a cancellation event because an email is never sent to our intermediary gmail account. We provide this test script as an alternative for your testing, however; in order to demonstrate full functionality you must use the production Procmail Filter.

```
# used for testing to see if attachment is populated
mailServer = smtplib.SMTP('smtp.gmail.com', 587)
mailServer.ehlo()
mailServer.starttls()
mailServer.ehlo()
mailServer.login("procmailtestsrelease@gmail.com", "gobeavers!")
mailServer.sendmail(email_from, new_email_to, new_email_obj.as_string())
mailServer.close()
```

Fig 22. The python email code

The above method of testing is demonstrated in our final video.