

BC-isit442 Class Overview And Structure

Document #PLPC-120048

Version 0.1

January 05, 2016

This Document is Available on-line at:

<http://mohsen.banan.1.byname.net/PLPC/120048>

Mohsen BANAN

Email: <http://mohsen.1.banan.byname.net/contact>

Contents

I	Message Sending – Submit Client/Server – Direct Injection	1
1	Messaging Injection	1
2	Submit Server	1
3	Submit Clients	1
II	Message Sending Logs Analysis	2
4	Where Are The Log Files	2
5	Tools For Analyzing Log Files	2
III	Class Activities	2
6	Blee And Emacs Proficiency	2
6.1	Using Emacs As An IDE	2
6.2	YASnippet	2
7	Bash Scripting – Mail Injector Exercise	3
7.1	MailInjector Development	3
7.2	MailInjector Evolution	3
7.3	MailInjector Functional Specification	3
7.4	MailInjector Design Specification Topics – Your Assignment	3
7.5	MailInjector Deeper Dive	4
7.6	Extending Our MailInjector	4
7.7	lcaMailfrontHosts.sh – Walkthrough	4
IV	Assignments	4
8	Review Of Last Week’s Assignments	4
9	Next Set Of Assignments	4

List of Figures

Part I

Message Sending – Submit Client/Server – Direct Injection

1 Messaging Injection

Messaging Injection

- Queuing For The Spooler
- Spooling
- Send Attempts
- Temporary Send Failures
- Permanent Send Failures
- Logging
- Non-Delivery Reports

2 Submit Server

- Frontmail
- Bellevue College's Submit SMTP Server

3 Submit Clients

Submit Clients

- qmail-inject – with smtproutes
- Various MUAs – Gnus
- ssmtp
- swaks
- blat (Windows)

Part II

Message Sending Logs Analysis

4 Where Are The Log Files

Where Are The Log Files

- /var/log/
- /var/log/qmail

5 Tools For Analyzing Log Files

Tools For Analyzing Log Files

- tai64nlocal

Part III

Class Activities

6 Blee And Emacs Proficiency

6.1 Using Emacs As An IDE

- Programming Language Modes – (Bash, Python, Lisp, C/C++, Html, CSS, JavaScript – TeX, Beamer)
- YASnippet <http://www.emacswiki.org/emacs-test/Yasnippet>
Useful Video: <http://www.youtube.com/watch?v=ZCGmZK4V7Sg>
- Dynamic Evaluation
- Linting and Style Enforcement
- Debugging

6.2 YASnippet

- Using Snippets
- Creating Your Own Snippet

7 Bash Scripting – Mail Injector Exercise

7.1 MailInjector Development

- Requirements / Functional Specification – (Mohsen)
- Design – (Students)
- Design Review – (Joint)
- Coding / Implementation – Very Evolutionary And Layered (Students)
- Verification – (Joint)
- Iteration – (Joint)

Keeping It All Together – Inside Of Your Script

7.2 MailInjector Evolution

- `cd /isit442`
- `mv mailInjector mailInjector.20160209`
- `cp -r -p /opt/public/osmt/bin/mailInjector .`

7.3 MailInjector Functional Specification

- Read In A File Structured As: `/opt/public/osmt/bin/mailInjector.data`
- 4 Comma Sepated Fields. Example: `mohsen.banan@bellevuecollege.edu,nn,Mohsen,Banan`
- Create A Message (Header and Body) Based On Fields (2,3,4)
- Send That Message To content of Field 1
- Produce A Report/Log Of All successful and failed injection/sends

7.4 MailInjector Design Specification Topics – Your Assignment

- How will you read the input file? `stdin`? parameter? argument?
- How will you parse the file?
- How will you create the message?
- How will you submit the message? (manual-entry “`qmail-inject`”)
- How will you specify the From: line?
- How will you specify the Envelope Address?
- How Will you determine the message injection was successful?
- How Will you determine the message was not delivered?
- How will you produce a report?

7.5 MailInjector Deeper Dive

- Top-Down Design, Bottom-Up Implementation
- Managing Complexity, Cohesion, Consistency – mailInjector.sh
- Panel.org Omni-Available Features
- Iim Omni-Available Features
- Bashing – Procedural vs Functional Vs Object-Oriented

7.6 Extending Our MailInjector

- New Templates – for example “nn”
- Multiple invocation of the same line – Or repetition of the full input
- Commenting the functions

7.7 IcaMailfrontHosts.sh – Walkthrough

- Disabling Of Other Components
- Creation Of The Run Commands
- Keys Generation
- Structure And Commonality Of Iim

Part IV

Assignments

8 Review Of Last Week’s Assignments

- Design Your MailInjector.sh Based On The Functional Spec
Do It However You Wish. Pseudo Code, High Level Bash, Flow Chart, English Description.
Include Error Handling and Data Analysis – How Should Input Be Delivered To It?
Consider Making It Parallel Vs Serial/Linear.
Consider Scaling

9 Next Set Of Assignments

- Google “qmail Log Analyzers”
- apt-cache search qmail | grep -i log

- qmailanalog
- man syslog (3)
- man logger (1)
- man splogger (8)
- Take mailInjector.sh and comment it out so it makes more sense.

Two Teams Of 3.

Team 1 uses swaks. Team 2 uses ssmtp

Customize the mailInjector.sh to hit lightly against Bellevue College with your student mail credentials.

References