IT441
Network Services Administration

Chris Kelly

cg.kelly2013@gmail.com

Goal of This Course

- The goal of this course is to teach you to automate the work of system and network administration using scripts written in the Perl scripting language.
- The goal of this first lecture is
  - to let you now how this course will be conducted and
  - to begin learning Perl.

Format of the Course

- *This is a combined* **lecture** *and* **lab** *course*
- I will speak for a time at the beginning of each class
- After that, you will spend the remainder of the class time working on class exercises and projects
    - I will be here to help you with any issues that may arise
    - **HINT:** When issues do arise, it to your great benefit to resolve them sooner, rather than later.
- Using a computer (your own or a machine in here), you will complete class exercises and project work

Format of the Course

- Class exercises will be graded, so it would be a mistake to skip them. Besides, they help develop skills.
  - Often, these exercises will be practice for techniques you will later use in projects.
  - Finally, to an extent, it is just a matter of doing them and not putting them off.
  - Exercises will have due dates
- You will need to read a number of chapters in *The Practice of System and Network Administration*, and submit *summaries* of what you have read

Format of the Course

- Much of the class will involve working on *projects* in which you build your knowledge of the Perl programming language, as well as applying it to system and network administration tasks.
- Project work will usually involve two components:
  - Perl code files
  - ***Reflections/reports***
- As previously mentioned, class exercises will serve as practice for a lot of what you do in projects, so please do not neglect them.
- Finally, there will be ***quizzes*** to ensure you are keeping up with the lectures and reading.

Assessment

- **Projects:** 50%
- **Final Project:** 20%
- **Class Exercises:** 15%
- **Quizzes:** 10%
- **Chapter Summaries:** 5%

Projects and Administrator's Log

- One of the most important things you can learn from this course, is the importance of keeping a written record of what you have done
- When you change a machine you administer (such as adding or updating a script) – or something significant happens on it – you should make a note in your admin log.
  - Changes to a machine's configuration can cause problems, that may not appear until *months* afterwards
  - If you forget what you changed and when, you will struggle figuring out what to do next
  - This is particularly important when you *solve a problem*
- The writing process also helps you to assimilate the knowledge...

Projects and Administrator's Log

- For this course you must keep an **Administrator's log** – which will consist of your memo files for each project. These will be in
  - project-specific subdirectories of your `it441/projects` directory
  - along with the other project-specific files
- The memos must be text files (`memo.txt`)
- When you are signed into Linux, the file paths will probably look something like this: `~/it441/projects/`
- The memos will consist of you documenting and reflecting on the process, as well as answering some discussion questions

Projects and Administrator's Log

- Each project's files will be due by a particular date and time – to be eligible for credit.
- You should make an entry in the log (i.e., in the memo file) for each day you work on the project
- Inside the `~/it441/projects/` directory, you will have a project specific subdirectories called `project_01`, `project_02`, and so forth.
- Directory names and file names **must** match those given in projects
- File permissions must also be correct for me to see and execute files

Course Textbooks

- The three textbooks for this course are quite different
- *Beginning Ubuntu LTS Server Administration* and *Ubuntu 16.04 LTS Server: Administration and Reference* describe how to set up and configure an Ubuntu Server
  - If you have the former from IT341, that will be fine.
  - Otherwise, you may acquire the latter, which is the new textbook for IT341, moving forward
- Either of these will help clarify many of the technical steps we go through during the course
- Do not neglect this reading!

Chapter Summaries

- *Beginning Perl* will introduce you to the use of the Perl programming language
- *The Practice of System and Network Administration* is written by veteran system administrators
- It contains practical advice for system administrators, gleaned from experience
- Reading this book will help you become a better system administrator
- Throughout the course, I'll assign chapters to read, along with *suggested* summary completion dates.

Chapter Summaries

- You will find the reading schedule on the course web page.
- We may have some discussion on these chapters, if we have time
- To make sure you have read this book, you must submit chapter summaries
- You will find the specifications for the chapter summaries on the course webpage
- You should get started on these ASAP!

Working on the Command Line and with Configuration Files

- Since all of you have taken IT 244 and IT 341, you know that the command line is a user hostile environment.
- Almost all system administration work done on Linux machines is done at the command line – or in text files
  - In Linux and Unix, almost all configuration information is stored in *text files*
  - All of the project work you do in this course, will be done at the command line
- You must be very careful about what you type at the command line
  - If you mistype or misspell a single character, your command will not work the way it is supposed to
  - As such, you must be *extremely careful* when changing these files

Working on the Command Line and with Configuration Files

- A single typo could cause some Linux service on your machine to fail
- Moreover, this class is very programming-heavy, which similarly requires considerable attention to detail and avoidance of typos
- Finally, you will also be expected to write code that is
  - Neatly-formatted and elegant
  - In a manner easily understandable to anyone reading the code.
- Material will be cumulative as the semester progresses.
- You will need to decide if you are able to invest the time and effort that this course will require.

Attendance

- At each class I'll take attendance
- I do this to:
  - Learn your names
  - Have a record
- Your attendance will not affect your grade directly
- However, if you find yourself struggling with the material and have not been coming to class, I'll be less sympathetic

Do You Have Enough Time to Do the Work for This Course?

- Many of you work, either part time or full time
- This cuts down on the time you have for class work
- You *should not* be taking this course if you do not have enough time to do all the work
- In this course, you will be writing programs in order to automate system administration tasks.
  - The command line is a user-hostile environment in general, and many students struggle with programming.
  - Programming requires a different way of thinking, more structured and algorithmic

Do You Have Enough Time to Do the Work for This Course?

- Moreover, excellent grades require excellent work from you.
  - Mediocre work will only earn mediocre grades.
  - So, you need to decide if you are willing to invest the time and effort needed for the grades you desire.
- If you sign up for more work than you can achieve in the time you have, you are cheating yourself
  - Many people in this country rush to get a degree, but haven't done enough work to digest the material
  - Those people invariably set themselves up for failure

Course Documents

- Everything I create for this class is made available online
- All of it can be accessed from the Class Page:

  `http://www.cs.umb.edu/~ckelly/teaching/it441`

- You should bookmark this page because the page will function as our syllabus, instead of a paper syllabus
- It is a lot of material, but you should at least get to know the layout
- The "Course Policies" section will give you a good idea of my rules and expectations.
- That section also contains some supplementary information you should check out.

Course Documents

- The schedule will feature links to class notes, along with reading assignments – *including your chapter summaries*
- The "Projects" section will feature descriptions of each project as they come up
- Many terms we encounter in this class can be found on the Definitions page: **`http://www.cs.umb.edu/~ckelly/teaching/it341/local_assets/files/common/data/linux/linux_sysadmin_definitions.html`**

Taking Notes

- Although I make my notes available in PDF form, I want to encourage you to take notes in class
- Studies have shown that students learn more when they take notes, even if they never look at their notes again
- Other studies have shown that the more activities and senses are engaged when you learn something, the greater your likelihood of remembering

Taking Notes

- Writing notes engages another part of your brain, which increases recollection
- All of you should take notes
- Probably the best practice would be for you to print the notes *before* coming to class.
- That way, you can write your own notes in the margins, along with any questions you may have.
- ***Note:*** *Sometimes PDF content may differ from slides as presented in class!*

Textbooks

- There are three textbooks for this course:
  - Beginning Perl (3rd edition), by James Lee (ISBN: 1430227931)
  - *Beginning Ubuntu LTS Server Administration* (2nd Edition) by Sander van Vugt., Apress, ISBN: 1430210826
  - *The Practice of System and Network Administration* (3rd Edition) by Limoncelli, Hogan and Chalup, Addison-Wesley Professional, ISBN: 0321919165

Cheating

- All students are expected to follow the University's Code of Student Conduct
- You will find this at **`http://www.umb.edu/life_on_campus/policies/community/code`**
- The Computer Science Department has the following policy on cheating
- You will be given a score of **zero** if you cheat on any assignment, quiz or test

Cheating

- If you cheat a second time you will receive an **F** in the course
- If you cheat a third time you can be **expelled** from the University
- I put a great deal of work into my courses, and I ask you to respect that work by not cheating.
- ***Important:*** *It is the student's responsibility to know what constitutes academic dishonesty – at this university and in this class. Lack of knowledge that something constitutes an academic honesty violation will not be accepted as a valid excuse.*

Courtesy and Decorum

- The following two items are matters of basic consideration:

1. When I am just coming into class and setting up, please hold your questions until I am finished and we begin. This way, I can give you my full attention and do a better job helping you.
2. When I or someone else is addressing the class, please put your conversations on hold. It is a matter of common courtesy, and the talking can be distracting for some of us.

Accommodations for Disabilities

- The school is legally obligated to try to accommodate students with disabilities
- If you have a disability you can get help from Ross Center for Disability Services
  - **Location:** Upper Level of the Campus Center, Room 211
  - **Phone:** 617-287-7430
  - **Web Site:** `https://www.umb.edu/academics/vpass/disability/`

Accommodations for Disabilities

- After you have discussed the matter with them, see me
  - They will usually draft a letter explaining any accommodations you should receive.
  - You should get this letter to me **ASAP!**
  - If you require extra time for an exam, then it is **<u>your</u>** responsibility to arrange for this at least a week in advance!
- Also, you may wish to check out the page containing my own notes:

`http://www.cs.umb.edu/~ckelly/teaching/common/data/disability.html`

Email

- All communication outside of class will be conducted through email
- For regular contact, we are going to use your **@umb.edu** or **@cs.umb.edu** email
- The first assignment will involve setting up your Linux account
- I will use that account when sending you a *personal* email concerning the class or any *class-wide* announcements outside of class. It is your responsibility to check both
- If I have sent you an email about something concerning the class, I'll assume that you have been given adequate notice

Contacting Me

- If you have a question, email me at **cg.kelly2013@gmail.com**
- Please be sure to:
  1) Use a descriptive, meaningful subject line
  2) Begin the subject with **IT441:**
- *Failing to include #2 is effectively the same as **not having sent** the e-mail at all!*
- Don't hesitate to contact me if you are stuck and/or need help with something.
- Others might be having the same issue!

Office Hours

- My office is S-3-130
- My official office hours are posted on the course web page
- You **do not** have to make a special appointment to see me during office hours – just drop in!
- If you need my help and cannot make it to office hours, contact me and we'll work something out

Types of Programming Languages

- There are two broad classes of programming languages:
  - Compiled languages
  - Scripting languages
- Compiled languages – like C, C++ and Java – are used in large programming projects
- In a compiled language, you create source code files – text files consisting of instructions in the language

Types of Programming Languages

- A source code file is written by programmers, but it cannot (in and of itself) be run on a computer
- In order to run the program you must create an executable file
- For example, here is the source code for a simple program written in C:

```
#include <stdio.h>
main()
{
  printf("Hello, world!\n");
}
```

Types of Programming Languages

- In order to create an executable file, I must run this file through a C compiler:

```
$ gcc -o hello hello.c
```

- This command creates the executable file hello , which I can run on the Unix machine like so:

```
$ ./hello
Hello, world!
```

Types of Programming Languages

- Compiled languages are designed for big programming projects, but they are overkill for simple system administration tasks
- Let's say we wanted to remove all files that meet the following criteria:
    - Within a directory and its subdirectories
    - Have the `.tmp` extension
    - Were created before a certain date

Types of Programming Languages

- You would not want to have to write a C program to do this
  - C is very finicky about syntax
  - It would probably give you a lot of compiler errors before you got a working program
- To write programs like this, a ***scripting language*** is preferable

Types of Programming Languages

- Scripting languages are not compiled
  - But they are still written by people in a programming language and cannot be run directly on the machine
  - Instead, they are run by an interpreter
- An ***interpreter*** is a program that understands the scripting language -- turning the source code file into actions performed on the machine

<u>Types of Programming Languages</u>

- Here are some scripting languages commonly in use today:
  - Perl
  - PHP
  - Python
  - Ruby
- We will be learning ***<u>Perl</u>***