EE5240: Computer Modeling of Power Systems

Dr. Gowtham

Director of Research Computing, IT
Adj. Asst. Professor, ECE and Physics

EERC B39 · (906) 487-4096 · g@mtu.edu · @sgowtham

Week #08: 2017-02-27, 2017-03-01 and 2017-03-03
Structured Programming Tools

Coding a reflection of one’s logic ... for the most part


© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/2
Problem Definition

※ Understand what is given

※ Understand what is expected

※ Understand what you have and what you don’t have for resources
Literature Search

- Your own personal and/or friends’/colleagues’ collection
- Computing literature
  - GitHub | Stack Overflow
- Object-oriented learning
Notes

- As detailed as possible
- Include date, time, and location
- Include hostname, and version of OS, software, compilers
- Hand-written, not just an electronic version


© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/5
Seymour Roger Cray (1925 – 1996): American electrical engineer, entrepreneur; founder, Cray Research
Use schematics/plots - a picture is still worth a thousand words

http://dilbert.com/strip/2009-03-07/

© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/7
Language of Choice

- Learn more than one if you can
- Know what best fits your research needs
- Realize that not every language does everything well

Examples: Scripting (BASH, PERL, Python), programming (C/C++, FORTRAN, Java, Julia, Mathematica, MATLAB), documentation ($\LaTeX$), database (SQL, Oracle), web design (CSS, HTML, PHP)

© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/8
Appearance: Write for computers people

- A sense of *love at first sight*
- Should look pretty, have a good and logical flow, and be useful
- Statements and modules in top-down/alphabetical order
Appearance

Write for computers people; useful but not pretty ugly

```
#include <stdio.h>
main(t, _, a)
char *a;
{return !0<t?t<3?main(-79,-13,a+main(-87,1_,
main(-86, 0, a+1 )+a)):1,t<_?main(t+1, _, a ):3,main ( -94, -27+t, a
)&t == 2 ?_<13 ?main ( 2, _+1, "%s %d %d\n" ):9:16:t<0?t<-72?main(_,
t,"@n'+,#'/*{}w+/w#cdnr/+,}r/*de)+,/*{++,/w{+,/w#q#n+,#/1,+/n{+\n,/+n+,#/q#n+,#+/k#/;\'r :'d*3,}{w+K w'K:'+}e#';dq#'l q#'+d'K#!'\n+k#;q#'r)eKK#{w'r}eKK{nl]'/#;q#'n'}{}#w'}(){}nl]'+/+n';d}rw' i;#}{n\nl]'}/n{n#' ;r#w'r nc{nl]'#/1,'+K {rw' iK{{[nl]'}/w#q#n w'w' iK{KK{nl}!/w%'l##w# ' i; :{nl]'/*{q#'ld;r'}{nlwb!/*de}'c \\{"nl'{-}{}rw'+'/;'+}#*}#nc,'#\nw]'+'/kd'+e}+;\n#'rdq#w! nr'/ ') }+}{rl}'{n''# }'# }'##(!/"
:t<-50?"==a ?putchar(a[31]):main(-65,_,a+1):main((a == '/')|t,_,a
+1 ):0?t?main ( 2, 2 , "%s":a==''/||main(0,main(-61,a, "!ek;dc \
i@bK'(q)-[w]*%n+r3#1,{}):\nuwloca-0;m .vpbks,fxntdCeghiry"),a+1);}
```
// ForkBomb.c
// C program to demonstrate the fork bomb with memory leak. Compilation takes
// less than one second on most modern hardware running Linux OS with GCC.
//
// Compilation and execution:
// gcc ForkBomb.c -o ForkBomb.x
// ./ForkBomb.x

// Headers
#include <stdlib.h>

// main()
int main() {

    while (1) {
        // Replicate and allocate 8 GB memory
        fork(); // Not a bad call by itself; infinite while loop makes it dangerous
double *ptr = (double *) malloc(1024 * 1024 * 1024 * sizeof(double));
    }

    // Indicate termination
    return 0;
}
// Factorial.c

// Computes factorial(n) where n is an integer (>=0) supplied by the user.
// Compilation/Execution takes about one second on most modern hardware
// running Linux OS with GCC.

// Compilation and execution:
// gcc Factorial.c -lm -o Factorial.x
// ./Factorial.x

// Headers
#include <stdio.h>

// Function declaration
int factorial(int n);

// main()
int main() {

// Variable declaration/initialization
int n = 0; // User-supplied number
int N = 1; // factorial(n)

// PRINT PROBLEM/PROGRAM STATEMENT
printf(" A non-negative integer: ");
scanf("%d", &n);

// VALIDATE USER INPUT
// Compute factorial and print result
N = factorial(n);
printf(" factorial(%d) = %d\n", n, N);

// Indicate termination
return 0;
}

// factorial()
int factorial(int n) {

// Variable declaration/initialization
int M = 1; // factorial(n)

// Compute the factorial
// factorial(0) or factorial(1) is 1
if (n == 0 || n == 1) {
    M = 1;
}

// Recursive approach for n > 1
if (n > 1) {
    M = n * factorial(n - 1);
}

// Return factorial to parent module
return M;
}
Communication

* Meaningful nomenclature and comments
  Variables, arrays, structures and functions

* Documentation with metrics
  OS, architecture, hardware, compiler, versions, compilation and execution instructions, time required to compile/run, input and output requirements

* Revision control system
  Keep a detailed track of development
Modularization  Divide n’ conquer

* Module/Sub-routine
  * Accomplishes recurring tasks efficiently
  * Reduces program size and makes debugging easier
  * Requires description and comments just like the main program

Augusta Ada King, Countess of Lovelace (1815 – 1852): English mathematician and writer
// sum_loop()
int sum_loop(int N) {

    // A sub-routine to compute the sum of first N integers for a given value of N
    // using a for loop.
    //
    // Usage:
    // sum = sum_loop(N);

    // Variable declaration and initialization
    int i = 0; // Loop index
    int sum = 0; // Sum of integers from 1 through N

    // Loop method
    for(i = 1; i <= N; i++) {
        sum = sum + i;
    }

    // Return the value of sum to the parent function/module
    return sum;
}
// sum_gauss()
int sum_gauss(int N) {

    // A sub-routine to compute the sum of first N integers for a given value of N
    // using Gauss' method.
    // Usage:
    // sum = sum_gauss(N);

    // Variable declaration and initialization
    int sum = 0;

    // Gauss method
    sum = N * (N + 1)/2;

    // Return the value of sum to the parent function/module
    return sum;
}

Johann Carl Friedrich Gauss (1777 – 1855): German mathematician
Augusta Ada King, Countess of Lovelace (1815 – 1852): English mathematician and writer
Testing

※ Check every line/step, and input/output
※ Be a devil’s advocate and check for extreme cases
※ Does the program do NOTHING when it is supposed to NOTHING?

Unit test
A method by which individual units of source code, sets of one or more program modules together with associated control data, usage procedures, and operating procedures are tested to determine whether they are fit for use. It helps find problems early, facilitates change, simplifies integration, improves documentation, and the code’s design.

Unit testing frameworks
© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/17
Testing

Regression test

A type of software testing that seeks to

1. uncover new bugs (i.e., regressions) in existing functional and non-functional areas of a system after some changes have been made

2. ensure aforementioned changes have not inadvertently introduced new bugs (or re-introduced previously fixed old bugs), often in a different part of the code

The cause for re-appearance of bugs is often a poor revision control practice (or lack of a formal one, such as Git). The cause for new bugs is often a poor design and/or a fragile fix to a problem (i.e., solution tested for a particular case but not in general).
Debugging

* Identify the bug and understand its solution

She was one of the first programmers of Harvard Mark I (1944), invented the first compiler for programming languages, and popularized the idea of machine-independent programming languages. US Navy guided-missile destroyer, USS Hopper, and Cray XE supercomputer at NERSC, Hopper, are named in her honor of her achievements.
Debugging

* Angry Spouse Bug
* Bloobug
* Bugfoot
* Common Law Feature
* Defensive Coding
* Heisenbug
* Higgs Bugson
* Hindenbug
* Hydra Code
* Jenga Code
* Loch Ness Monster Bug
* Lorem Ipsum Bug
* Ninja Comments
* Reality 101
* Unicorn
* Yoda Conditions

http://blog.codinghorror.com/new-programming-jargon/
Ian Cummings (itcummin@mtu.edu), a PhD candidate in ECE, has written a script/program that reads through a MATLAB .m file, and uses the comments to prepare LaTeX documentation.
Optimization/Profiling

* Modifying the code to run more efficiently

Premature optimization
Act of letting performance considerations affect the code's design.

Design → Code → Debug → Optimize
It is better to design, then code from the design, and then profile or benchmark the resulting code to identify which parts should/can be optimized.

A simple and elegant design is often easier to optimize, and profiling may reveal unexpected performance problems that would be hidden behind the curtain of premature optimization.
Integrated Development Environment (IDE)

- Source code editor
- Syntax highlighting
- Intelligent code completion
- Build automation tools
- Debugger and profiler
- Compiler and/or interpreter
- Support for revision control system
- Object-oriented programming features
Additional References

https://www.youtube.com/watch?v=QdVFvsCWxrA
Additional References

* The Art Of Computer Programming, vol. 1-4A

* The Idea Factory: Bell Labs And The Great Age Of American Innovation
  J. Gertner; Penguin Press (2012)

* The Design Of Everyday Things
  D. Norman; Basic Press (2013)

* Doxygen Official website | GitHub
  Automatic generation of documentation from source code

* Michigan Tech Multiliteracies Center (Walker Arts Building #107)
Additional References

* IDEs

CLion (C/C++) | MATLAB | PyCharm | RStudio
Vi(m): #1, #2, #3, #4, #5, #6, #7, #8, #9

* Twitter

@AdviceToWriters | @Doxygen | @Grammarly | @PurdueWLab
@WritersDigest | @WritersRelief | @WritingCom | @Writing_Tips
@inside_R | @MATLAB | @RBloggers | @RLangTip | @ROpenSci
@RProgLangRR | @RStudio | @RStudioTips | @R_Programming
Revision Control System

Travel back and forth between revisions

Instructor’s Dissertation  How it looked without a formal revision control system

[sgowtham@feynman Dissertation]$ ls
20070924.0 20071025.0 20071123.0 20071203.0 20071216.0 20080114.0
20070924.1 20071026.000 20071124.0 20071204.0 20071217.0 20080114.bw
20070925.0 20071030.0 20071125.0 20071205.000 20071218.0 20080114.color
20070927.0 20071030.1 20071126.0 20071211.0 20071219.0 20080121.bw
20070928.0 20071119.0 20071127.0 20071211.1 20071220.0 20080121.color
20071002.0 20071120.0 20071128.0 20071212.0 20071220.0 20080122.color
20071022.0 20071121.0 20071129.0 20071213.0 20080127.0 20080122.color
20071023.0 20071122.0 20071129.0 20071214.0 20080128.0 20080129.0

[sgowtham@feynman Dissertation]$ cd 20080122.color
[sgowtham@feynman 20080122.color]$ ls
Abstract.tex 	Chapter6.tex 	MTUPhDThesis.sty
Abstract.txt 	Chapter7.bib 	MTUPhDThesis.sty.0
Acknowledgements.tex 	Chapter7.tex 	MyThesis.bib
Appendix.tex 	Dedication.tex 	MyThesis.dvi
Beowulf_Cluster.tex 	Future_Work.tex 	MyThesis.tex
Bibliography.tex 	Graphs 	Nano_Bio_Physics.bib
Chapter1.bib 	Images 	Nano_Bio_Physics.tex
Chapter1.tex 	Index.tex 	nextpage.sty
Chapter2.bib 	Introduction.bib 	PublishedPapers
Chapter2.tex 	Introduction.tex 	README.PLEASE
Chapter3.bib 	ListOfFigures.tex 
TableOfContents.tex
Chapter3.tex 	ListOfPublications.bib 
Theoretical_Details.bib
Chapter4.bib 	ListOfPublications.tex 
Theoretical_Details.tex
Chapter4.tex 	ListOfTables.tex 
TOC.pdf
Chapter5.bib 	Makefile 
TOC.tex
Chapter5.bib 	Metal_Oxide_Clusters.bib
Chapter6.bib 	Metal_Oxide_Clusters.tex
Did not have to spend time learning something new near graduation

Spent a lot of time incorporating edits from advisor and advisory committee members, and between versions

An incomplete sentence, and missed out on thanking six good friends (and their parents) in the final printed copy as a result of picking an incorrect version to continue editing

Lifelong shame of being inept and ungrateful
Instructor’s Dissertation: How it would have looked with a formal revision control system
Git

A distributed RCS with an emphasis on speed, data integrity, and support for distributed, non-linear workflows, and single/multiple users working on single/multiple projects.

Every working copy is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server.

Potential applications

Systems administration, software development, manuscript preparation, event planning, etc.

http://git-scm.com
Linus Benedict Torvalds (1965 – present): Finnish American software engineer
Git and GitHub

GitHub, world’s largest code host

A safe, secure and social web-based hosting service for software development projects that use Git revision control system. GitHub’s copy is usually treated as the most trustworthy repository.

* The learning curve can be steep
* A form of data backup that keeps track of the workflow
* Easily move back and forth between revisions
* A readily available portfolio for potential employers
* Saves space, time, $, and creates opportunities

http://github.com
.gitignore

* Every Git repository should have one at its very top level

* List of files, folders and file types that should **not** be in the repository
  
  * OS- and language-specific temporary files
  
  * System files and symbolic links
  
  * Program executables and other binary files
  
  * Files with large data sets and/or sensitive information

* A class of entities can be specified with wild card characters
Git Commit History

Textual output

```
git log --pretty=format:"%h - %an, %ad : %s"
```

Graphical output

```
gource --hide dirnames,filenames --seconds-per-day 0.1 \ 
   --auto-skip-seconds 1 -1280x720 -o - | \ 
   ffmpeg -y -r 60 -f image2pipe -vcodec ppm -i - \ 
   -vcodec libx264 -preset ultrafast -pix_fmt \ 
   yuv420p -crf 1 -threads 0 -bf 0 PROJECT_NAME.mp4
```

Git and MATLAB R2014b and beyond


© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/34
SnailGit on iTunes App Store (the free version supports one while the paid version supports many repositories).
**Additional References**

* Git

  Reference | Book | Videos | External links
  Tagging | Forking | Branching and merging

* Git – structuring commit messages:  #1, #2, #3

* GitHub

  Interactive tutorial | Cheat sheet | Online training | Desktop version

* Twitter

  @GitHub | @GitHubEducation | @GitHubStatus

* **Open a GitHub.com account** (optional)

  Try to keep GitHub username same as Michigan Tech ISO username
  If you already have an account, there’s no need to open a new one
Debugging With MATLAB

The art of finding and fixing mistakes

http://dilbert.com/strip/1995-11-14/

© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/37
Commonly Used Techniques In Debugging Programs

* Taking detailed notes
* Using `printf()` (or equivalent) statements
* Logging with Standard Logging Frameworks using the debug flag
* Smarter text editors: emacs | gedit | Sublime Text | vim
* Free and open source tools: ddd | eclipse | gdb | valgrind
* Commercial tools: IBM Rational Purify | IDB | MATLAB | pgdbg

Debuggers are usually the last line of defense

There is no substitute for good programming etiquette OR taking detailed notes. Messages from debuggers often look cryptic to an untrained eye, and might require some effort to understand them.
Graphical Techniques  Approximate value of $\pi$

\[
\pi_{\text{computed}} = \left[ \frac{2\sqrt{2}}{9801} \sum_{n=0}^{\infty} \frac{(4n)! (1103 + 26390n)}{(n!)^4 396^{4n}} \right]^{-1}
\]

\[
\epsilon = |\pi_{\text{known}} - \pi_{\text{computed}}|
\]

\[
\pi_{\text{known}} = 3.141592653589793
\]

\[
\delta = 10^{-15}
\]

$\delta$ is the accepted value of zero (also known as the tolerance).
Graphical Techniques Approximate value of $\pi$

- Possible workflow
  - Use double-precision (i.e., `format long`)
  - Identify the constants (i.e., scaling factor, $2\sqrt{2}/9801$)
  - Simplify the core within the loop
  - Estimate the error associated with computed value of $\pi$
  - Loop should end when error is less than a given tolerance, $\delta$
  - Display the results
## Advanced Techniques

**MATLAB Command Window**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbstop</td>
<td>Set breakpoints for debugging</td>
</tr>
<tr>
<td>dbstatus</td>
<td>List all breakpoints</td>
</tr>
<tr>
<td>dbstep</td>
<td>Execute next executable line from the current breakpoint</td>
</tr>
<tr>
<td>dbcont</td>
<td>Resume execution</td>
</tr>
<tr>
<td>dbclear</td>
<td>Remove breakpoints</td>
</tr>
<tr>
<td>dbtype</td>
<td>Display file with line numbers</td>
</tr>
<tr>
<td>dbstack</td>
<td>Function call stack</td>
</tr>
<tr>
<td>keyboard</td>
<td>Input from keyboard</td>
</tr>
<tr>
<td>dbquit</td>
<td>Quit debugging mode</td>
</tr>
</tbody>
</table>


© Gowtham 2017

EE5240: Computer Modeling of Power Systems · 08/41
Potential Pitfalls

Highlight reel of some of my biggest blunders since 2002

http://dilbert.com/strips/comic/1999-09-14/

© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/42
This won’t happen to me Syndrome

- Getting enough sleep/rest
- Budgeting time and resources
- Taking detailed notes
- Using `printf()` (or equivalent) statements
- Describing the workflow to someone else
- Having someone else look at the code
- Understanding what the language can and cannot do
- Integrating more than one language into the workflow

Code samples in AdditionalMaterial/CRTErrors/ are good candidates for this failed experiment.
Blind Copy, Compilation and Execution

- Read through the borrowed code
- Check if your hardware meets the criteria

This pitfall can often cause hardware damage beyond repair.
# Variable Sizes and Limits

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(signed) char</td>
<td>-128</td>
<td>127</td>
<td>1</td>
</tr>
<tr>
<td>unsigned char</td>
<td>0</td>
<td>255</td>
<td>1</td>
</tr>
<tr>
<td>(signed) int</td>
<td>-2147483648</td>
<td>2147483647</td>
<td>4</td>
</tr>
<tr>
<td>unsigned int</td>
<td>0</td>
<td>4294967295</td>
<td>4</td>
</tr>
<tr>
<td>(unsigned) short int</td>
<td>-32768</td>
<td>32767</td>
<td>2</td>
</tr>
<tr>
<td>unsigned short int</td>
<td>0</td>
<td>65535</td>
<td>2</td>
</tr>
<tr>
<td>(signed) long int</td>
<td>-9223372036854775808</td>
<td>9223372036854775807</td>
<td>8</td>
</tr>
<tr>
<td>unsigned long int</td>
<td>0</td>
<td>18446744073709551615</td>
<td>8</td>
</tr>
<tr>
<td>(signed) long long int</td>
<td>-9223372036854775808</td>
<td>9223372036854775807</td>
<td>8</td>
</tr>
<tr>
<td>unsigned long long int</td>
<td>0</td>
<td>18446744073709551615</td>
<td>8</td>
</tr>
<tr>
<td>float (6 digits)</td>
<td>1.17549e-38</td>
<td>3.40282e+38</td>
<td>4</td>
</tr>
<tr>
<td>double (15 digits)</td>
<td>2.22507e-308</td>
<td>1.79769e+308</td>
<td>8</td>
</tr>
<tr>
<td>long double (18 digits)</td>
<td>3.3621e-4932</td>
<td>1.18973e+4932</td>
<td>16</td>
</tr>
</tbody>
</table>

If a variable is assigned a value higher (or lower) than its defined upper (or lower) limit, then the value stored is the maximum (or minimum) value.
Scope of Variables

Observe the value of \( x \) before, within and after the `while` loop.

<table>
<thead>
<tr>
<th>Location</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before the while loop begins</td>
<td>3.1415</td>
</tr>
<tr>
<td>Within the while loop</td>
<td></td>
</tr>
<tr>
<td># 01</td>
<td>1.0101</td>
</tr>
<tr>
<td># 02</td>
<td>2.0202</td>
</tr>
<tr>
<td># 03</td>
<td>3.0303</td>
</tr>
<tr>
<td># 04</td>
<td>4.0404</td>
</tr>
<tr>
<td># 05</td>
<td>5.0505</td>
</tr>
<tr>
<td>After the while loop ends</td>
<td>3.1415</td>
</tr>
</tbody>
</table>

[sgowtham@feynman JoFE]$ ./ScopeOfVariables.x

[sgowtham@feynman JoFE]$
Uninitialized Variables

Observe the warning issued by the compiler

```
Without the `printf()` statements within `sum_uninitialized` function, it's quite tough to find this error.
```
Once the program produces meaningful result, comment the `printf()` statements used for debugging purposes.
## Assignment vs Equality

A single `=` represents the assignment operator. For e.g., `x = 42` means **assign the value 42 to variable** `x`.

A double `==` is used to check equality. For e.g., `if x == y` means **check if** `x` **has the same value as** `y`.

## Yoda condition

Checking if a constant equals the variable instead of the other way.

```java
if (42 == x) {
    DO SOMETHING
}
```
Single- vs Double-Precision

Adding one to the sum of one part in a billion a billion times

\[
\left[ \sum_{n=1}^{10^9} 10^{-9} \right] + 1 = ?
\]

Adding one part in a billion a billion times to one

- \( x \) [single precision] : 1.031250
- \( y \) [double precision] : 2.000000
Single- vs Double-Precision

Adding the sum of one part in a billion a billion times to one

\[ 1 + \sum_{n=1}^{10^9} 10^{-9} = ? \]

Adding one to one part in a billion a billion times

\[ x \text{ [single precision]} : 1.000000 \]
\[ y \text{ [double precision]} : 2.000000 \]
Addition is Not Associative

\[ a + b + c, \ (a + b) + c, \ \text{and} \ a + (b + c) \ \text{may not be same} \]
Subtracting Nearly Equal Numbers

Functional derivative definition implies \( f'(x) \) gets better as \( h \to 0 \)

\[
f'(x) = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h}
\]

With \( f(x) = \sin(x) \),

\[
f'_{\text{Functional}}(x) = \lim_{h \to 0} \frac{\sin(x + h) - \sin(x)}{h}
\]

\[
f'_{\text{Analytical}}(x) = \cos(x)
\]
### Subtracting Nearly Equal Numbers

What's the value of $h$ that minimizes the error?

<table>
<thead>
<tr>
<th>$h$</th>
<th>Functional derivative</th>
<th>Analytical derivative</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e+00</td>
<td>0.0678264420177852</td>
<td>0.5403023058681398</td>
<td>5e-01</td>
</tr>
<tr>
<td>1e-02</td>
<td>0.5360859810118690</td>
<td>0.5403023058681398</td>
<td>4e-03</td>
</tr>
<tr>
<td>1e-04</td>
<td>0.5402602314186211</td>
<td>0.5403023058681398</td>
<td>4e-05</td>
</tr>
<tr>
<td>1e-06</td>
<td>0.5403018851213304</td>
<td>0.5403023058681398</td>
<td>4e-07</td>
</tr>
<tr>
<td>1e-08</td>
<td>0.5403023028982545</td>
<td>0.5403023058681398</td>
<td>3e-09</td>
</tr>
<tr>
<td>1e-10</td>
<td>0.5403022473871033</td>
<td>0.5403023058681398</td>
<td>6e-08</td>
</tr>
<tr>
<td>1e-12</td>
<td>0.5403455460850637</td>
<td>0.5403023058681398</td>
<td>4e-05</td>
</tr>
<tr>
<td>1e-14</td>
<td>0.5440092820663267</td>
<td>0.5403023058681398</td>
<td>4e-03</td>
</tr>
<tr>
<td>1e-16</td>
<td>0.00000000000000000</td>
<td>0.5403023058681398</td>
<td>5e-01</td>
</tr>
<tr>
<td>1e-18</td>
<td>0.00000000000000000</td>
<td>0.5403023058681398</td>
<td>5e-01</td>
</tr>
<tr>
<td>1e-20</td>
<td>0.00000000000000000</td>
<td>0.5403023058681398</td>
<td>5e-01</td>
</tr>
</tbody>
</table>
Integer Division, Type Upgrade and Casting

\[ a \text{ and } b \text{ are integers}; \ p \text{ and } q \text{ are double-precision.}\]

One can invoke double-precision in MATLAB using the command `format long`.

Different programming languages treat variable declaration differently.
Zero is Not Really Zero

Tolerance, $\delta$, is the tolerable/accepted value of zero. It can be used to check if the value of two variables is identical.

$\delta$ can change from one problem (or project) to another.
### Zero Costs Space and Time

Storing every double-precision element requires 800 bytes
Storing only non-zero double-precision elements requires 80 bytes

Matrix in which most elements are zero is **sparse**, and one in which most elements are non-zero is **dense**.

\[
A = \begin{pmatrix}
0 & 0 & 0 & 0 & a_{14} & 0 & 0 & 0 & 0 & 0 \\
0 & a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & a_{55} & 0 & 0 & a_{58} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a_{66} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{74} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & a_{92} & 0 & 0 & 0 & 0 & a_{87} & 0 & 0 \\
0 & 0 & 0 & a_{92} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]
Row-Major Language  C, C++, Python

\[ A = \begin{pmatrix} 
  a_{11} & a_{12} & \cdots & a_{1n} \\
  a_{21} & a_{22} & \cdots & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{m1} & a_{m2} & \cdots & a_{mn} 
\end{pmatrix} \]

* A, laid out in linear fashion, would look like

\[ a_{11} \; a_{12} \; \ldots \; a_{1n} \; a_{21} \; a_{22} \; \ldots \; a_{2n} \; \ldots \; a_{m1} \; a_{m2} \; \ldots \; a_{mn} \]

* To loop through the array in above order
  * First, loop over rows
  * Next, loop over columns
Column-Major Language \ FORTRAN, MATLAB, Octave, R

\[
A = \begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{pmatrix}
\]

* \( A \), laid out in linear fashion, would look like

\[
a_{11} \ a_{21} \ \ldots \ \ a_{m1} \ a_{12} \ a_{22} \ \ldots \ a_{m2} \ \ldots \ a_{1n} \ a_{2n} \ \ldots \ a_{mn}
\]

* To loop through the array in above order
  * First, loop over columns
  * Next, loop over rows
Memory Pre-Allocation and Memory Leak

**Pre-allocation**
The act of checking the availability of required amount of memory, and if available, reserving it to store entities at the beginning of a program.

** Leak**
The act of not releasing (i.e., freeing up) the memory that is no longer necessary. Leaks often occur when a memory allocations are incorrectly managed by a program and/or when an entity stored in memory cannot be accessed by the running code.

MemoryPreAllocation.c is in AdditionalMaterial/JoFE/.
ForkBomb.c discussed previously demonstrated both these aspects but in a very dangerous fashion.
Things To Try

※ Division by zero
※ Square root of a negative number
※ Array population/manipulation starting with 0 as the array index
※ Generate the Fibonacci sequence

\[ F(n) = F(n-1) + F(n-2) \]

Use \( F(0) = 0 \) and \( F(1) = 1 \) for \( n = 2, 3, 4, \ldots \)

※ A multi-file project where the master file calls one or more dependent files but not all dependent files are in the same (or designated) folder as the master file
gdb  The Gnu Debugger

* Installed by default on most Linux machines
* Supports C, C++, FORTRAN, Java and Python
* Historical recall (with arrow keys) and auto-completion (with tab)
* More information about a specific topic can be accessed via help

Compiling, running and debugging a C program

gcc -Wall -g -pg PROGRAM.c -lm -o PROGRAM.x
./PROGRAM.x

gdb -q ./PROGRAM.x
run
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>run</code></td>
<td>Run <code>PROGRAM.x</code></td>
</tr>
<tr>
<td><code>kill</code></td>
<td>Stop executing <code>PROGRAM.x</code></td>
</tr>
<tr>
<td><code>help</code></td>
<td>Get help on debugger commands</td>
</tr>
<tr>
<td><code>list</code></td>
<td>List the source code in <code>PROGRAM.c</code>, 10 lines at a time</td>
</tr>
<tr>
<td><code>list M,N</code></td>
<td>List the source code between lines <code>M</code> and <code>N</code></td>
</tr>
<tr>
<td><code>break M</code></td>
<td>Pause the execution at line <code>M</code> (i.e., set a breakpoint)</td>
</tr>
<tr>
<td><code>continue</code></td>
<td>Continue the execution</td>
</tr>
<tr>
<td><code>delete M</code></td>
<td>Delete the pause at line <code>M</code> (i.e., remove a breakpoint)</td>
</tr>
<tr>
<td><code>step</code></td>
<td>Execute the current line in source code but stop before the next line</td>
</tr>
<tr>
<td><code>next</code></td>
<td>Execute the next line in source code</td>
</tr>
<tr>
<td><code>print EXPR</code></td>
<td>Print the value of expression <code>EXPR</code></td>
</tr>
<tr>
<td><code>quit</code></td>
<td>Exit <code>gdb</code></td>
</tr>
</tbody>
</table>

Observe the similarity between these and MATLAB commands.
Additional References

- An Introduction To Fast Format
- Logging Frameworks
  - Boost (C++), Pantheios (C/C++). SLF4J (Java), etc.
- The Science Of Debugging
  - M. Telles, Y. Hsieh; Coriolis Technology Press (2001)
- Twitter
  - @AnoushNajarian | @HadleyWickham
Before We Meet On Friday

- Locate an IT-managed lab with a Linux workstation
  
  http://www.mtu.edu/it/services/computer-labs/
  
  Note down the location (building, floor, room, etc.)

- Verify that you can log into one such workstation using your Michigan Tech ISO credentials

- Change your default login shell to `/bin/bash`
  
  * Log into https://mylogin.mtu.edu/

  * Click on My Profile tab

  * Select `/bin/bash` from NIS Shell dropdown list

  * Click on Submit
Linux

The free and open source operating system

I'll sell you the rights to use Linux for one million dollars.

After the first month, you only need to pay for every upgrade.

It sounds too good to be true.

It's not as if I'm giving it away for free.

Linux is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of, and a prime example for concept and practice of, free and open source software development and distribution. The underlying source code may be used, modified, and distributed — commercially or non-commercially — by anyone under licenses such as the GNU GPL.

Linux is user friendly but ... 

It is picky as to who its friends are, and often very unforgiving of mistakes. It prefers friends to be committed to mindful practice, and be sensitive to case, space, and other weird characters.

Linus Benedict Torvalds (1965 – present): Finnish American software engineer
Linux gymnasium

* colossus.it.mtu.edu and guardian.it.mtu.edu
  * Intel Xeon X5675 3.07 GHz, 24 CPU cores, 96 GB RAM
  * Accessible for all from anywhere via SSH using a Terminal
* Linux workstation in a campus lab/office
  * May not be as powerful as colossus.it or guardian.it
  * May not be directly accessible from off-campus

Just so you know

All IT-managed workstations, unless explicitly indicated otherwise, run RHEL 7.x and will mount your campus home directory.
Develop a personalized yet consistent scheme

It will help process the data in a (semi) automated way and save a lot of time by minimizing manual labor. Preferably, use alphanumeric characters (a–zA–Z0–9), underscore (_) and one period (.) in file/folder.

Parsing other special characters, !@#$%^&*();:–?/\=+,. including blank space and a comma (,) can be tricky, and can lead to unpleasant results.

The scheme can be extended to include naming variables, arrays, and other data structures during software development.
Additional References

* FOSS 101: Essentials of Free and Open Source Software
  * Free and online course from Michigan Tech
  * 10 total modules with chain-like dependency
  * Each module has 10 untimed yes/no-like tasks and unlimited attempts (+ a *module completion badge* to show off in your Canvas profile)
  * Attempt to work through the first six modules
  * Contact Dr. Gowtham when in need of help with these tasks
  * Dr. Mork will observe of our progress
Additional References

* POSIX Compliance | GNU General Public License
* Linux | The Linux Command Line | The Command Line Crash Course
* Red Hat Enterprise Linux (RHEL) | CentOS Project | Fedora
* Vi(m) editor: Interactive Tutorial | Reference
* BASH Scripting/Programming: Introduction | Beginners | Advanced
* Twitter
  @CLIMagic | @Linux | @LinuxFoundation | @Linux_Tips | @RegExTip
  @MasteringVim | @UNIXToolTip | @UseVim | @VimLinks | @VimTips
Linux Server Setup
A bird’s eye view of general HOW TOs, DOs and DON’Ts


© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/72
Installation

* Type #1: Complete
  * Install all packages and services, and then disable unnecessary services
  * Time to install/update packages can be very long
  * Installation usually takes care of package dependency
  * Forgetting to disable/enable critical services can be a show-stopper

* Type #2: Selective
  * Install only necessary packages and services
  * Time to install/update packages can be short
  * Resolving new package dependency can be non-trivial/time consuming
Services

Inexhaustive list

* Database (MySQL, PostgreSQL, Oracle)
* DHCP and DNS
* Email (IMAP and POP3; SMTP)
* LDAP (directory services; e.g., *stalker net* at Michigan Tech)
* Load Balancing (balance incoming traffic amongst different servers)
* Print
* Programming and Scripting
* SSH (remote access) and FTP (files and folders)
* Web (Apache, Tomcat)
Security  Firewall

* Quickest way to protect a new installation
* Should be implemented before the server goes online
* Change the default username/password for the server/services
* Keeps the door (i.e., access) to a service open to necessary sources

From Dr. Gowtham’s Journal of Failed Experiments

Office workstation, feynman.it.mtu.edu, had 193 failed login attempts as root within the first 10 minutes of its installation.

Brand new installation of Raspberry Pi 3 was being used by someone else as a hopping point and a playstation server within 24 hours of going online (forgot to change the default username and password).
Security: Ownership and permission

* Ownership of files, folders and other entities
  * **user**: the user who created the entity (i.e., the owner)
  * **group**: the group of users associated with the entity
  * **others**: everybody else

* Permissions for files, folders and other entities
  * **read**: read the entity (numerical value: 4)
  * **write**: write to/modify the entity (numerical value: 2)
  * **execute**: run the entity (numerical value: 1)

Run `ls -l` in a Terminal and observe the first column of output
Identification

★ Name

★ Needs a fully qualified domain name (FQDN) to be on the network
   
   Jim’s Foodmart
   feynman.it.mtu.edu or superior.research.mtu.edu

★ Can have aliases/nicknames for easier (local) identification
   
   Jim’s or feynman or superior

★ Number (i.e., IP address)

★ Needs an IPv4/IPv6 address that corresponds to the FQDN
   
   300 Pearl Street, Houghton, MI 49931
   141.219.41.21 or 141.219.92.69
IP Address  
*Ain't no place like 127.0.0.1!*

- **Special purposes**
  - Loopback and diagnostic functions: 127.0.0.0 – 127.255.255.255
  - Multicast groups: 224.0.0.0 – 239.255.255.255
  - Future use and R&D: 240.0.0.0 – 254.255.255.254

- **Private/Internal network (e.g., router at home, office, etc.)**
  - Class A: 10.0.0.0 – 10.255.255.255
  - Class B: 172.16.0.0 – 172.31.255.255
  - Class C: 192.168.0.0 – 192.168.255.255

- **Public/External network (everything else)**
Things To Try

Finding IP address given the hostname (DNS lookup)

Observe that the host command returns two DIFFERENT IP addresses for the same hostname.
Things To Try

Finding hostname given the IP address (reverse DNS lookup)

```
[fehnman] 10:29:59 ~]$ host 141.219.41.21
21.41.219.141.in-addr.arpa domain name pointer fehnman.it.mtu.edu.
[fehnman] 10:30:00 ~]$ host 216.58.192.238
238.192.58.216.in-addr.arpa domain name pointer ord30s26-in-f235.100.net.
238.192.58.216.in-addr.arpa domain name pointer ord30s26-in-f14.100.net.
[fehnman] 10:31:29 ~]$ 
```
Things To Try

Checking if a server is online (pinging)

```
[feynman 10:41:19 ~]$ ping -c 10 google.com
PING google.com (216.58.192.206) 56(84) bytes of data.
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=1 ttl=55 time=10.8 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=2 ttl=55 time=10.8 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=3 ttl=55 time=10.8 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=4 ttl=55 time=10.6 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=5 ttl=55 time=10.8 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=6 ttl=55 time=10.5 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=7 ttl=55 time=10.6 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=8 ttl=55 time=10.6 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=9 ttl=55 time=10.8 ms
64 bytes from ord30s25-in-f14.1e100.net (216.58.192.206): icmp_seq=10 ttl=55 time=10.7 ms

--- google.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 1809ms
rtt min/avg/max/mdev = 10.512/10.743/10.895/0.136 ms
[feynman 10:41:22 ~]$  
```

A really useful tool to include in (automated) workflows – such as transferring data to or from a remote server.

© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/81
Things To Try

Checking the path of a packet of information

```
[feynman 10:47:20 ~]$ traceroute mtu.edu
traceroute to mtu.edu (141.219.70.117), 30 hops max, 60 byte packets
1  mx400-07-001-staff-it-z16.tc.mtu.edu (141.219.40.1) 0.268 ms 0.267 ms 0.256 ms
2  mtu.edu (141.219.70.117) 0.170 ms 0.459 ms 0.446 ms
[feynman 10:47:21 ~]$ traceroute msi.umn.edu
traceroute to msi.umn.edu (160.94.221.133), 30 hops max, 60 byte packets
1  mx400-07-001-staff-it-z16.tc.mtu.edu (141.219.40.1) 2.724 ms 2.669 ms 2.623 ms
2  mx80-07-002.tc.mtu.edu (141.219.183.97) 0.330 ms 0.296 ms 0.241 ms
3  xe-0-0-3.hgtm-cor-mtu.mich.net (207.75.40.9) 3.169 ms 3.155 ms 3.113 ms
4  irbx70.pwrs-corpowers.mich.net (198.108.22.77) 4.063 ms 4.138 ms 4.118 ms
5  ae0x18.nw-ch13.mich.net (198.108.22.34) 10.332 ms 10.303 ms 10.350 ms
21.230 ms
7  telecomb-gr-01-1-hu-0-9-0-0-0-1.northernlights.gigapop.net (146.57.252.213) 21.321 ms 21.138 ms 2
1.185 ms
8  telecomb-br-01-po5-2002.northernlights.gigapop.net (146.57.255.5) 21.608 ms 21.674 ms 21.600 ms
9  telecomb-br-01-po5-2002.northernlights.gigapop.net (146.57.255.5) 21.608 ms 21.674 ms 21.600 ms
10  msi-temp.umn.edu (160.94.221.133) 21.633 ms 21.431 ms 21.350 ms
11  ***
12  ***
13  ***
14  ***
15  ***
16  **
17  oit-lbw-ltmselfmgd-727.claoit.umn.edu (160.94.140.22) 660.990 ms !H 219.005 ms !H 218.924 ms !H
[feynman 10:47:58 ~]$
```
Things To Try

Uptime and summary of network connections

```
[feynman 10:58:10 ~]$ uptime
10:58:12 up 16 days, 2:33, 5 users, load average: 0.27, 0.20, 0.16
[feynman 10:58:12 ~]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address            State
tcp   0      0 feynman.it.mtu.edu:47676 linuxplesk7.openho:http TIME_WAIT
tcp   0      0 feynman.it.mtu.edu:47666 linuxplesk7.openho:http ESTABLISHED
tcp   0      0 feynman.it.mtu.edu:47670 linuxplesk7.openho:http ESTABLISHED
tcp   0      0 feynman.it.mtu.edu:47678 sweater-227-237.rov:53295 ESTABLISHED
udp   0      0 feynman.it.mtu.edu:45918 superior-login1.res:ssh ESTABLISHED
udp   0      0 feynman.it.mtu.edu:33546 mail.tecmint.com:http TIME_WAIT
tcp   0      0 feynman.it.mtu.edu:49200 ord30s25-in-f196:https ESTABLISHED
udp   0      0 feynman.it.mtu.edu:42530 ord30s25-in-f206:https ESTABLISHED
tcp   0      0 feynman.it.mtu.edu:47668 linuxplesk7.openho:http ESTABLISHED
udp   0      0 feynman.it.mtu.edu:42574 ord30s25-in-f206:https ESTABLISHED
udp   0      0 feynman.it.mtu.edu:47674 linuxplesk7.openho:http ESTABLISHED
udp   0      0 feynman.it.mtu.edu:68440 ord30s25-in-f198:https ESTABLISHED
udp   0      0 feynman.it.mtu.edu:34104 ord30s21-in-f141:https ESTABLISHED
udp   0      0 feynman.it.mtu.edu:56840 lx-in-f189.1e100:https ESTABLISHED
udp   0      0 feynman.it.mtu.edu:47672 linuxplesk7.openho:http ESTABLISHED
udp   0      0 feynman.it.mtu.edu:33550 mail.tecmint.com:http TIME_WAIT
udp   0      0 feynman.it.mtu.edu:46182 ord30s25-in-f197:https ESTABLISHED
udp   0      0 feynman.it.mtu.edu:33548 mail.tecmint.com:http TIME_WAIT
udp   0      0 feynman.it.mtu.edu:43338 up2.com:nttp ESTABLISHED
```
Things To Try

Securely shred a file

```
[feynman 11:07:57 ~]$ shred -n 10 -v SecretFile.dat
shred: SecretFile.dat: pass 1/10 (random)...
shred: SecretFile.dat: pass 2/10 (555555)...
shred: SecretFile.dat: pass 3/10 (aaaaaa)...
shred: SecretFile.dat: pass 4/10 (924924)...
shred: SecretFile.dat: pass 5/10 (492492)...
shred: SecretFile.dat: pass 6/10 (random)...
shred: SecretFile.dat: pass 7/10 (666666)...
shred: SecretFile.dat: pass 8/10 (000000)...
shred: SecretFile.dat: pass 9/10 (fffffff)...
shred: SecretFile.dat: pass 10/10 (random)...
[feynman 11:08:12 ~]$
```
Things To Try

List every process from every user
### Things To Try

List every process from a given user

```bash
[feynman 11:33:16 ~]$ top -b -n 1 -u sgowtham
top - 11:33:19 up 16 days,  3:08,  7 users,  load average: 0.08, 0.28,  0.29
Tasks: 265 total,  1 running, 256 sleeping,  8 stopped,  0 zombie
%Cpu(s):  1.1 us,  0.3 sy,  0.0 ni,  98.5 id,  0.0wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 85777552 total, 50674580 free,  4180720 used, 10922252 buff/cache
KiB Swap: 33030140 total, 33030140 free,  0 used. 60542956 avall Mem

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR S</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+ COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>9921</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>562748</td>
<td>16328</td>
<td>5544 S</td>
<td>6.2</td>
<td>0.0</td>
<td>3:13.80 caribou</td>
</tr>
<tr>
<td>1977</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>144988</td>
<td>2332</td>
<td>1068 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.13 sshd</td>
</tr>
<tr>
<td>1993</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>120784</td>
<td>7492</td>
<td>1744 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.07 bash</td>
</tr>
<tr>
<td>2057</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>79032</td>
<td>4232</td>
<td>3276 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.12 ssh</td>
</tr>
<tr>
<td>3013</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>386136</td>
<td>5984</td>
<td>3155 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.03 gvfsd-http</td>
</tr>
<tr>
<td>4034</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>53756</td>
<td>1008</td>
<td>420 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.13 ssh-agent</td>
</tr>
<tr>
<td>4352</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>120864</td>
<td>7676</td>
<td>1849 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.40 bash</td>
</tr>
<tr>
<td>6248</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>149664</td>
<td>5568</td>
<td>2612 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.06 vim</td>
</tr>
<tr>
<td>9543</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>387940</td>
<td>4272</td>
<td>3336 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:12.08 gnome-keyr+</td>
</tr>
<tr>
<td>9550</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>663732</td>
<td>8472</td>
<td>6316 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:09.95 gnome-sess+</td>
</tr>
<tr>
<td>9557</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>13944</td>
<td>596</td>
<td>452 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.00 dbus-launch</td>
</tr>
<tr>
<td>9558</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>102160</td>
<td>3032</td>
<td>1192 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:29.14 dbus-daemon</td>
</tr>
<tr>
<td>9626</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>380488</td>
<td>3532</td>
<td>2829 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.19 gvfsd</td>
</tr>
<tr>
<td>9631</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>434936</td>
<td>5632</td>
<td>2900 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.03 gvfsd-daemon</td>
</tr>
<tr>
<td>9718</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>52864</td>
<td>568</td>
<td>0 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:05.18 ssh-agent</td>
</tr>
<tr>
<td>9752</td>
<td>sgowtham</td>
<td>20</td>
<td>0</td>
<td>337912</td>
<td>5516</td>
<td>2876 S</td>
<td>0.0</td>
<td>0.0</td>
<td>0:00.00 at-spi-bus+</td>
</tr>
</tbody>
</table>
```

© Gowtham 2017
EE5240: Computer Modeling of Power Systems · 08/86
Automation

* Manual feed (run one command at a time)

```bash
last > file_01.tmp
sed "/^\s*$\)/d" file_01.tmp > file_02.tmp
awk '{ print $1 }' file_02.tmp > file_03.tmp
sort file_03.tmp > file_04.tmp
uniq -c file_04.tmp > file_05.tmp
sort -nr file_05.tmp
```

* Piping

The act of using output of one command as the input for the next command

```bash
last | sed "/^\s*$\)/d" | awk '{ print $1 }' | sort | \
uniq -c | sort -nr
```
Automation

* Function and script
  
  A (portable) entity with a list of commands to accomplish a task

* Cron job
  
  A service that's useful to run a command (or a script or any other program) at a designated time (say, at 3 am on every Saturday) without needing user initiation/intervention.

Why is it a generally not a good idea to schedule something to run at 1 am or 2 am?
Additional References

* A Guide From Newbies To System Administrator

* Explore the built-in manual page for a given command (press q to exit out of it)

  man man
  man host
  man ping
  man traceroute
  man mkdir
  man netstat

* Explore the built-in manual page for a random command (useful for discovering new commands)

  man $(ls /bin | shuf | head -1)
Got questions?

EERC B39 · (906) 487-4096 · g@mtu.edu · @sgowtham