

COMP2101 - Administrative Shell Scripting

---

# Scripting Overview

---

---

# What is a Shell Script?

- Commands can be given to a computer by typing them into an interactive command interpreter program, commonly called a shell
- In this course, we will define a shell script as one or more shell commands saved in a file intended to be executed as a single program
- bash and Powershell are interactive command interpreters commonly used in the Linux and Windows environments as shells
- Many interactive command interpreters can be run using saved scripts
- Commands in scripts are generally executed the same as if they were typed interactively into the running command interpreter program

---

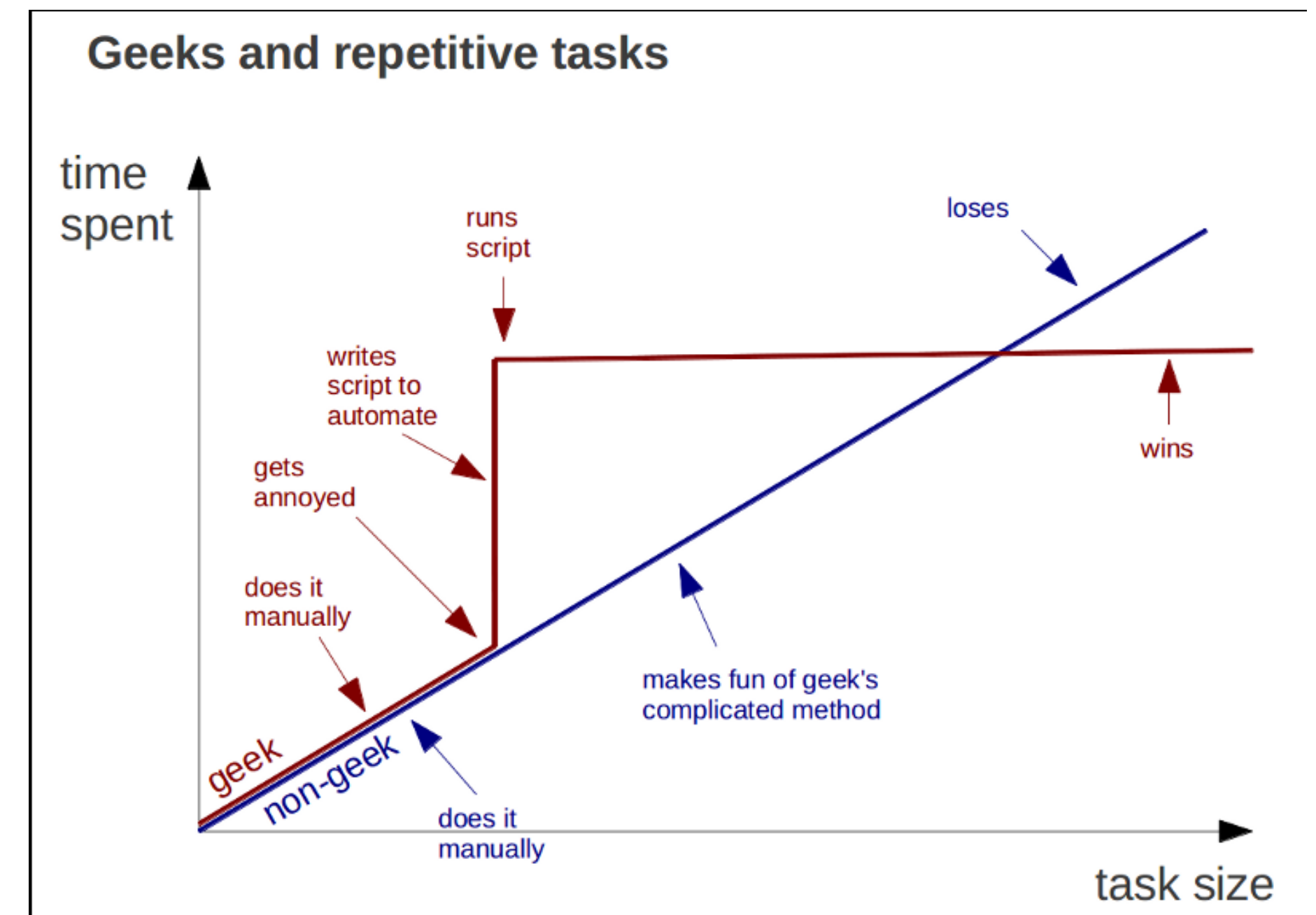
# Shell Scripting Goals

- Saving commands into a script allows us to:
  - Create tools to separate a script user from the details of a task they are performing
  - Communicate task requirements and status to a user in a way the user can understand
  - Perform checks to ensure the task is properly run and deal with problems that may arise in carrying out the task as requested by the user
- Using shell scripts to achieve these goals provides financial, process, and user empowerment benefits



# Financial Benefits of Scripting

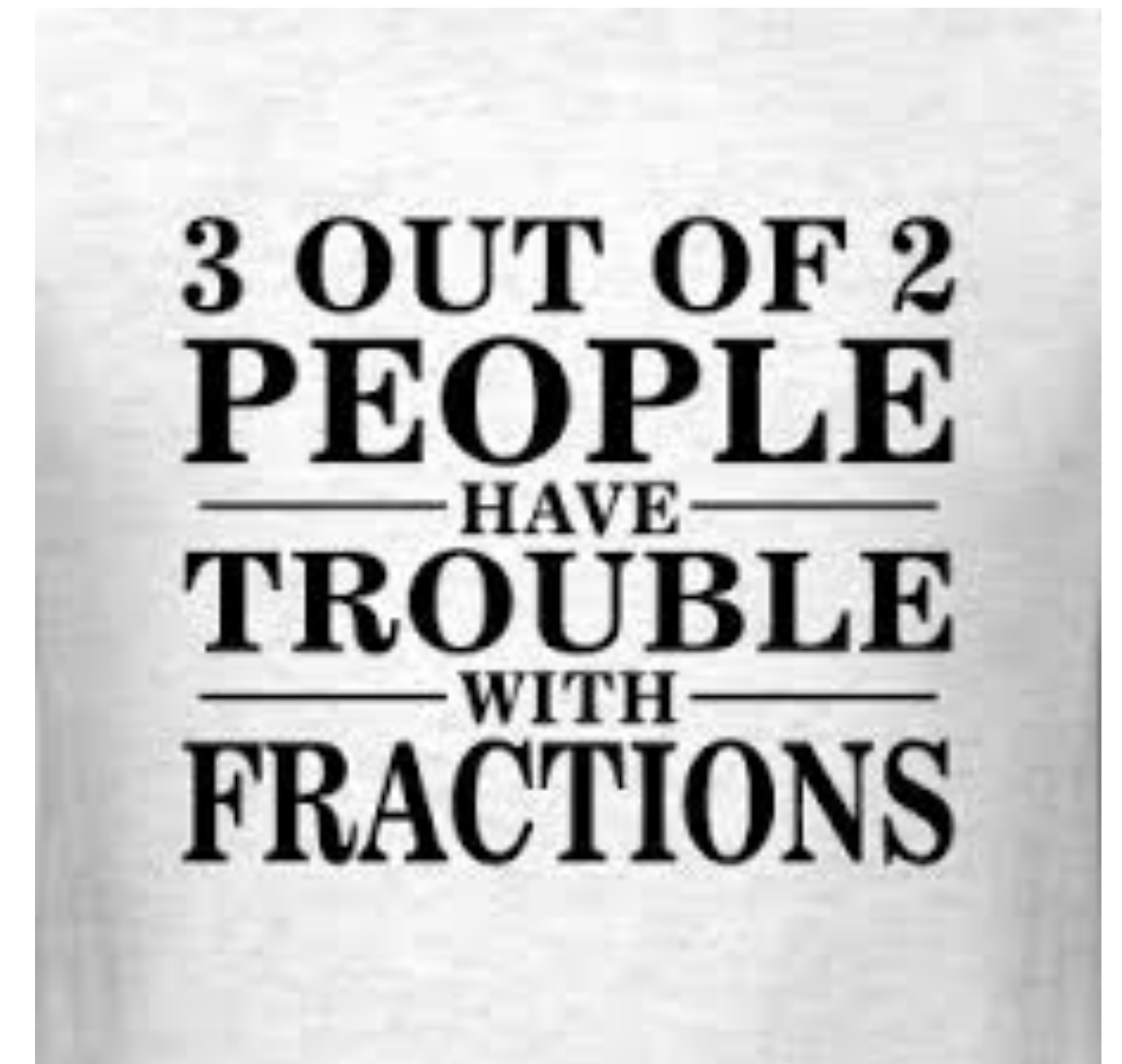
- The largest financial benefits over time of scripting are realized by automation of repetitive tasks
- Common things to automate for cost reduction:
  - making edits to large amounts of data or to data which repeats over time
  - evaluating, summarizing, and/or reporting on data content
  - making changes to a number of files, devices, machines
  - parallelizing tasks
- Scripting can increase amount of work done in a given amount of time and either eliminates labour freeing up staff for other tasks or enables use of unskilled labour



---

# Process Reliability Benefits of Scripting

- Processes that involve using a computer can be made more reliable by reduction or avoidance of human error
  - errors in command entry can be reduced or eliminated by not having users enter the actual commands
  - user input can be verified as valid before allowing a program to use that input
  - process steps cannot be skipped or scrambled because scripts run all commands in the script, in the correct order, every time
- The most seasoned shell veteran is susceptible to making mistakes or typos, just like the most junior user



# User Empowerment Benefits of Scripting

- Scripts as tools allow the simplification of complex tasks providing the opportunity to have people use them to perform tasks without extensive training
- Sequences of commands are reduced to single commands from the user's perspective
- Multiple platform and scenario handling can be accommodated in the script avoiding or reducing the need for users to be aware that different computing environments may function in different ways
- Errors and unexpected results handling can be incorporated providing resiliency in the process being automated



---

# Scripts vs. Compiled Applications

- Use of shell scripts instead of, or in addition to, compiled applications
  - provides deployment flexibility and ease
  - reduces development time and resource requirements
  - eases change management
  - extends capabilities of users of compiled applications

---

# When to Script

- Any time a computer is used to achieve a task, it is worth considering automating some or all of that task
- Characteristics of tasks that are indicators of when automation would be beneficial:
  - performing a repetitive task on multiple items of data or on multiple computers
  - applying data transformation or evaluation/summarization to a large quantity of data
  - working with complex command option combinations or command sequences
  - ensuring computing activities and results conform to business policy
  - flexibility in deployment and ease of program functionality change is important
- Scripting gives tools to users so they can perform tasks they do not necessarily have the knowledge to directly perform, scripts must take this into account and be able to insulate users from the details they are not expected to be responsible for



---

# Shells Used In This Course

- In Linux, the command shell is typically bash, current version is 5 but you may encounter 4 or even 3
- Windows PowerShell 5.1 is the current and final version of Windows PowerShell and is only usable on Windows
- PowerShell Core 7 is available for both Windows and non-Windows platforms, although it is missing most commands and much of its purpose unless you use it on Windows
- Shells evolve over time and new features get added, existing features can change, and features may be removed

Shell version can be very important to the correct functioning of your scripts!

---

# Lab Requirements

- We will be using bash version 5 on Linux in this course, the bash manual page is the primary documentation, WSL is not a suitable Linux environment for learning administrative shell scripting - native, VM, or cloud-based Linux systems will be required
- We will be using Windows PowerShell 5.1 in this course, the built-in help and Microsoft developer web site are our documentation resources - to know why, see [Microsoft's explanation of the difference between PowerShell and Powershell](#) (not a typo)
- Ubuntu Linux is used for the bash shell portion of the course, and Windows 10 is used for the Powershell portion of the course
- Use of other Linux distros, or other Windows versions is at the discretion of the student, however it will be the student's responsibility to research and work around compatibility problems encountered

---

# Script Preservation and Versioning

- It is required in this course to use [github.com](https://github.com) to save your files and provide version history for this course - we will be configuring it in our first class, and you will be marked on the files saved to your [github.com](https://github.com) course repository throughout the semester
- Each student will be required to create a repository named COMP2101 specifically for this course, and maintain it for the duration of the course
- The repository can be private, must be shared with the instructor in order to receive marks for lab work
- We use github in a trivial way, students wishing to go beyond the trivial use we make of it might want to read through [Pro GIT - a free ebook with a wealth of information about using git and github](#)

---

# Development System Setup

*Software can be chaotic, but we make it work*



*Expert*

Trying Stuff  
Until it Works

O RLY?

*The Practical Developer*  
*@ThePracticalDev*

- Ubuntu Desktop
- Github account and repo
- Git command
- Shellcheck command
- hello world script