

# Objectives

- Lab 1 Feedback
- Why Linux/UNIX?
- Lab 2

# Lab 1 Feedback

- Lookin' good
  - Good commenting
  - Good naming of variables
  - Good test cases

# Lab 1 Feedback: Variable Names

- Use descriptive variable names
  - Example: for atomic weight, the input from the user about the number of hydrogen atoms:
    - num\_h
- Begin with lowercase letter for typical variables
- All caps for constants
  - Variables that don't change during program execution

# Lab 1 Feedback: Testing

- Demonstrate that your program works using **several different** test cases when there is user-input
  - Use well-known test cases, when applicable
  - For example with F-->C, use 32F, 212F
- Don't show me testing from when you were developing the program
  - Only after you've done all testing/debugging
  - Restart IDLE shell to get rid of previous testing results

# Lab 1 Feedback: Data Types

- Don't need to use constructor/converter if variable is of correct data type
  - `x = raw_input("...")`
    - x is a string
  - `print "You entered" + x + "."`
    - Note don't need to use `str(x)` because x is already a string

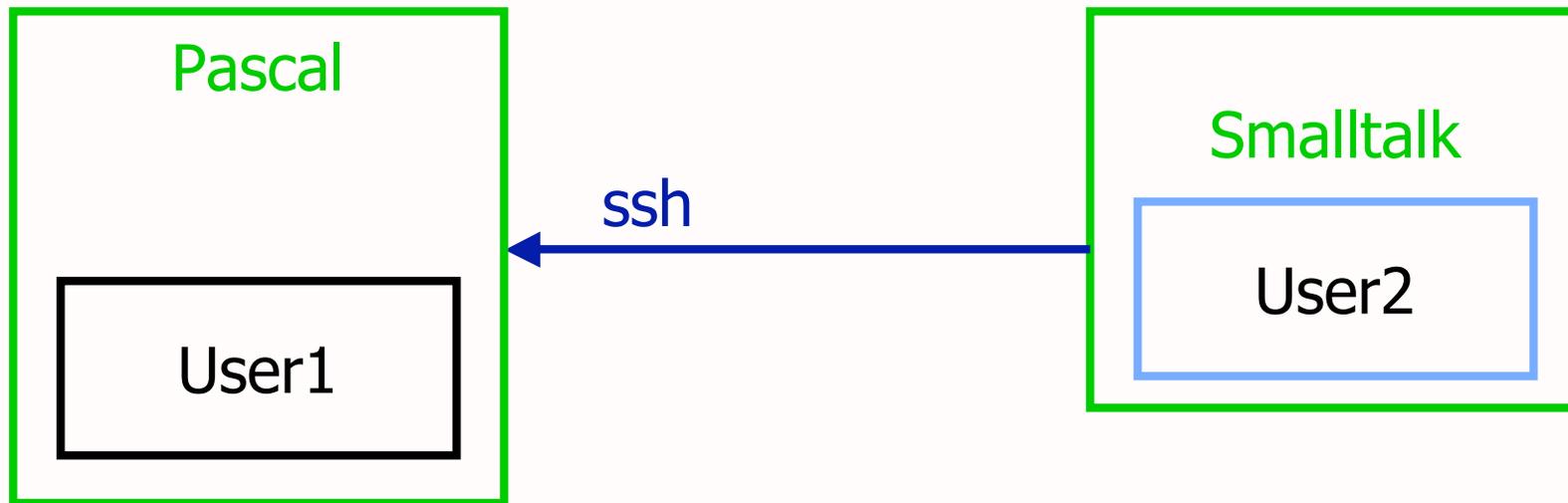
# Why Linux/UNIX?

- Depending on who you ask, you'll get different answers

# Why Linux/UNIX?

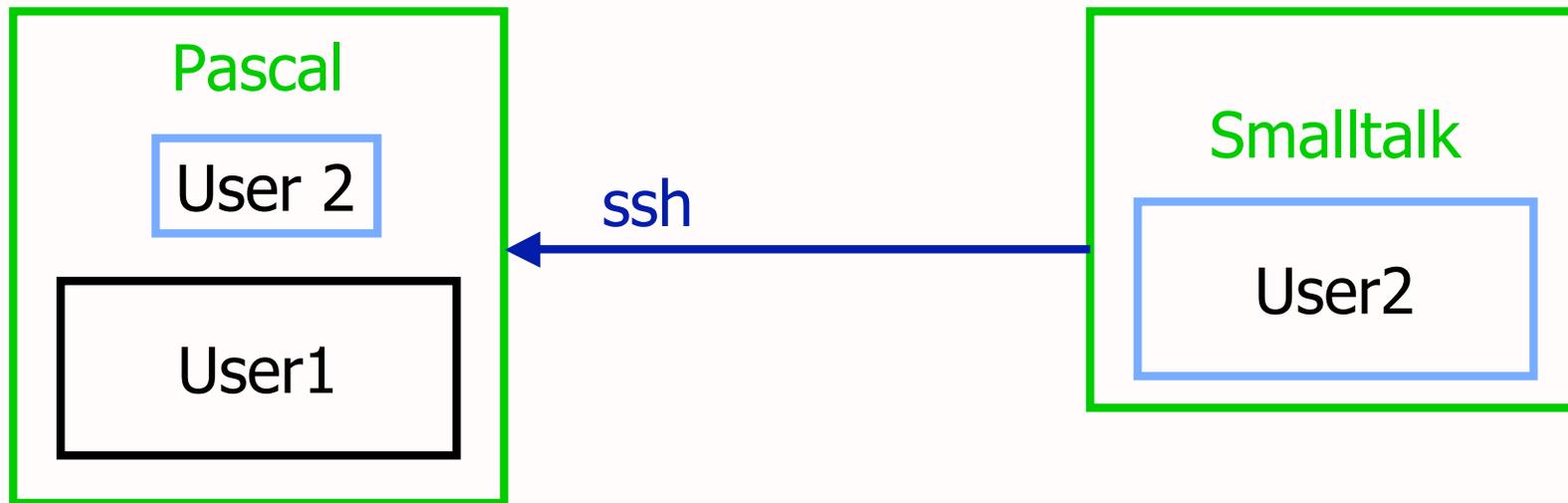
- Operating system reasons
  - Stability -- reboot less often
  - Efficiency (requires fewer hardware resources)
  - More secure (fewer viruses)
- User operations efficiency
  - Reduce amount of time that user's hands leave keyboard
    - Reduce mouse clicks
- *Real* multi-user capabilities
  - More than one user can use the machine's resources

# Using Machine Resources Remotely



- User1 physically sitting at machine named **pascal**
- User2 physically sitting at machine named **smalltalk**
- User 2 remotely logs into pascal using ssh
  - **SSH: secure shell protocol**

# Using Machine Resources Remotely



- User2 can execute programs *remotely* on **pascal** machine from command line
  - Can't use desktop manager
- User1 and User2 share pascal's resources

# Benefits of Remote Access

- Utilize more resources
  - Compute partial answers on different machines
  - Combine partial answers into final answer
  - Fewer idle CPU cycles
    - Take advantage of other machine's resources
- Applications:
  - Distributed and parallel processing
  - Scientific computing

# Lab 2

- Focus: formatting output, for loops
- Practice using Linux remote access
- Less help on stuff you should know
- Less help on initial problem-solving steps
  - Simulate the problem/solution on paper
  - Write out your algorithm on paper