

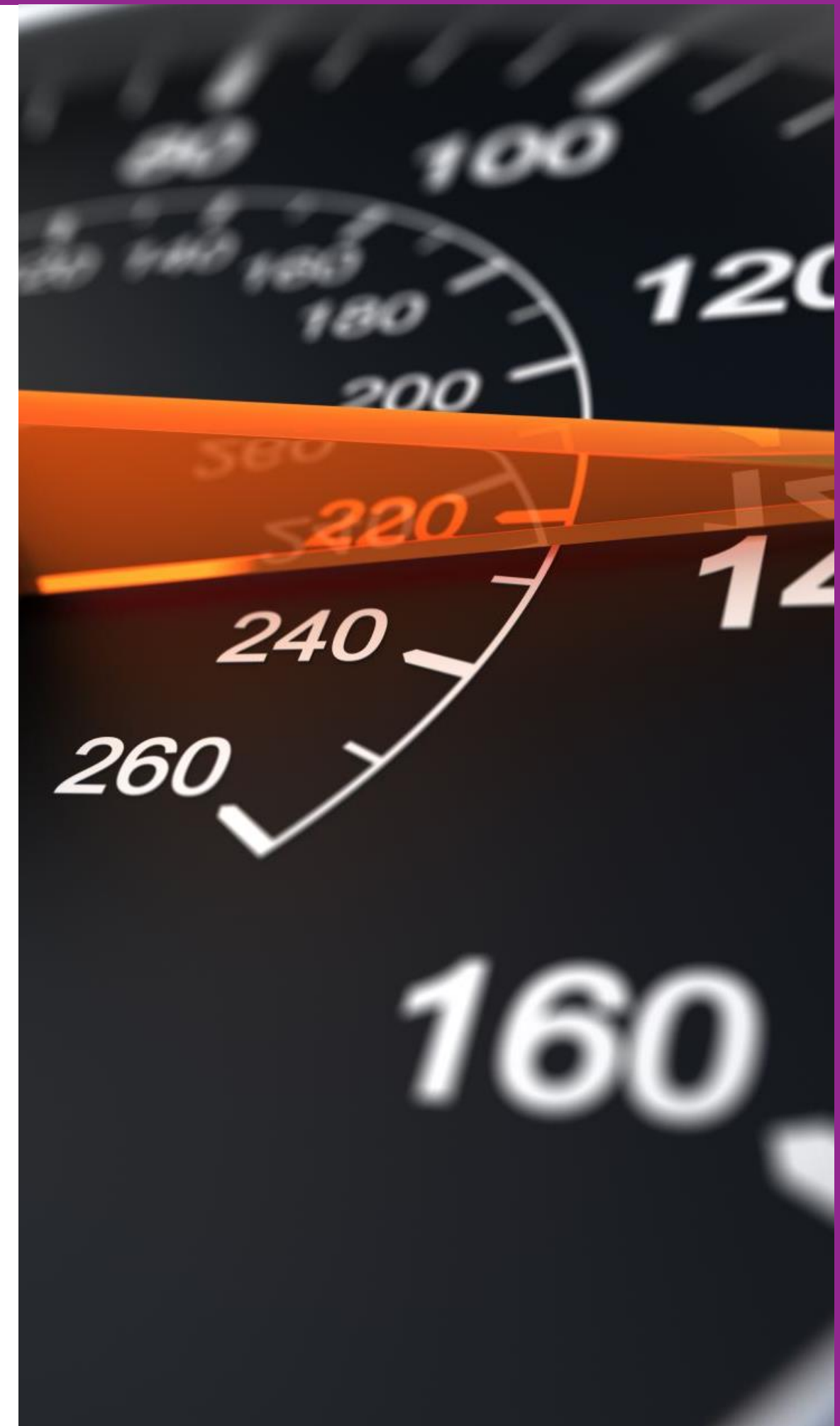
COMMAND LINE INTERFACE

Command Line Interface

- Go into any explorer window and type "cmd.exe"
- It will open a command prompt (in the current folder)
- This is also called a console or terminal window
- The command prompt is a text-based interfaces for interacting with the OS
- I recommend reading the following for some background:
- https://en.wikipedia.org/wiki/Command-line_interface

Command Prompt Shortcut Keys

- **"tab"** - auto-complete a file name in local directory
- **"tab"** - next auto-complete option
- **"shift-tab"** - previous auto-complete option
- **"up arrow"** - show previous command in history
- **"down arrow"** to show next command in history
- **"enter"** - copy selected to clipboard
- Right-click to paste from clipboard



Common Useful Commands

- **cd** – change directory, or current directory
- **call** – calls a batch command
- **echo** – outputs command-line arguments
- **more** – outputs input interactively screen by screen
- **type** – outputs the contents of a file
- **dir** – lists the contents of directory
- **find** – finds pattern in standard input
- **sort** – sort standard input line by line
- **findstr** – finds a string in files
- **fc** – file compare
- **set** – retrieves or assigns environment variables
- **assoc** – list file extension associations
- **tasklist** – gets the list of active processes
- **help** – displays help on a command
- **cls** – clears the command prompt buffer
- **cmd** – opens a new command prompt
- **where** – finds a program or file
- **xcopy** – copies files and directory trees

Most Commands are Executables

Most commands are console programs

- Also called [character-mode applications](#)
- See documentation on [consoles](#)

Accept input in three primary ways

- Standard input
- Command-line arguments
- Environment variables

Provide output via three primary ways

- Standard output
- Standard error
- Exit codes

Standard In and Standard Output

- A stream is a read-only or write-only sequence of data
- Usually either characters or bytes
- [StreamWriter](#) writes characters in a particular encoding
- [StreamReader](#) reads characters in a particular encoding
- [Console.WriteLine\(\)](#) writes to standard output
- [Console.ReadLine\(\)](#) reads from standard input
- Streams are opened or closed
- Usually unidirectional (can't back up unless buffered)
- Will be covered in more depth in later sessions

Filters and Pipelines

- Any program that processes an input stream and produces an output stream
- Usually a console application (no GUI)
- Most commands are filters
- Multiple filters can be chained together in a pipeline using the pipe ('|') operator

Command Line Arguments

- A set of values following a command name
- They are space delimited
- File paths that have spaces are surrounded in quotes
- Quoted file paths are treated as one input
- Switches (options) are command-line options by convention written as “/option”
- Options control program behavior, or how other arguments are interpreted
- Varies between program to program, and operating system to operating system
- Almost all support “/?”

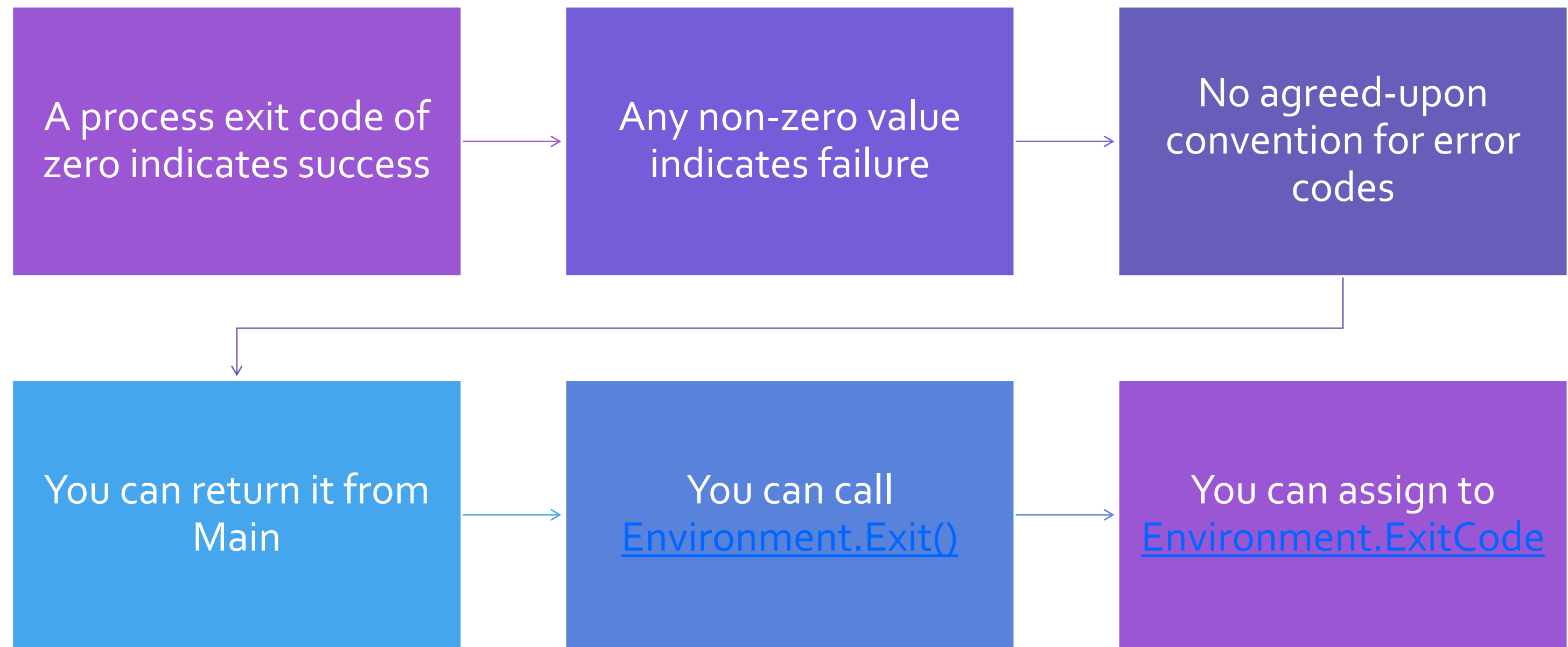
Retrieving Command Line Arguments

Passed to the main function as an array of strings

Can also use
[Environment.GetCommandLineArgs\(\)](#)

Note: GetCommandLineArgs() passes the executable path as the first argument

Process Exit Codes



Redirect (<, >) and Pipe (|)



`command < file` - Redirects the file contents into the standard input of a command



`command > file` – Redirects the standard output of a command into a file

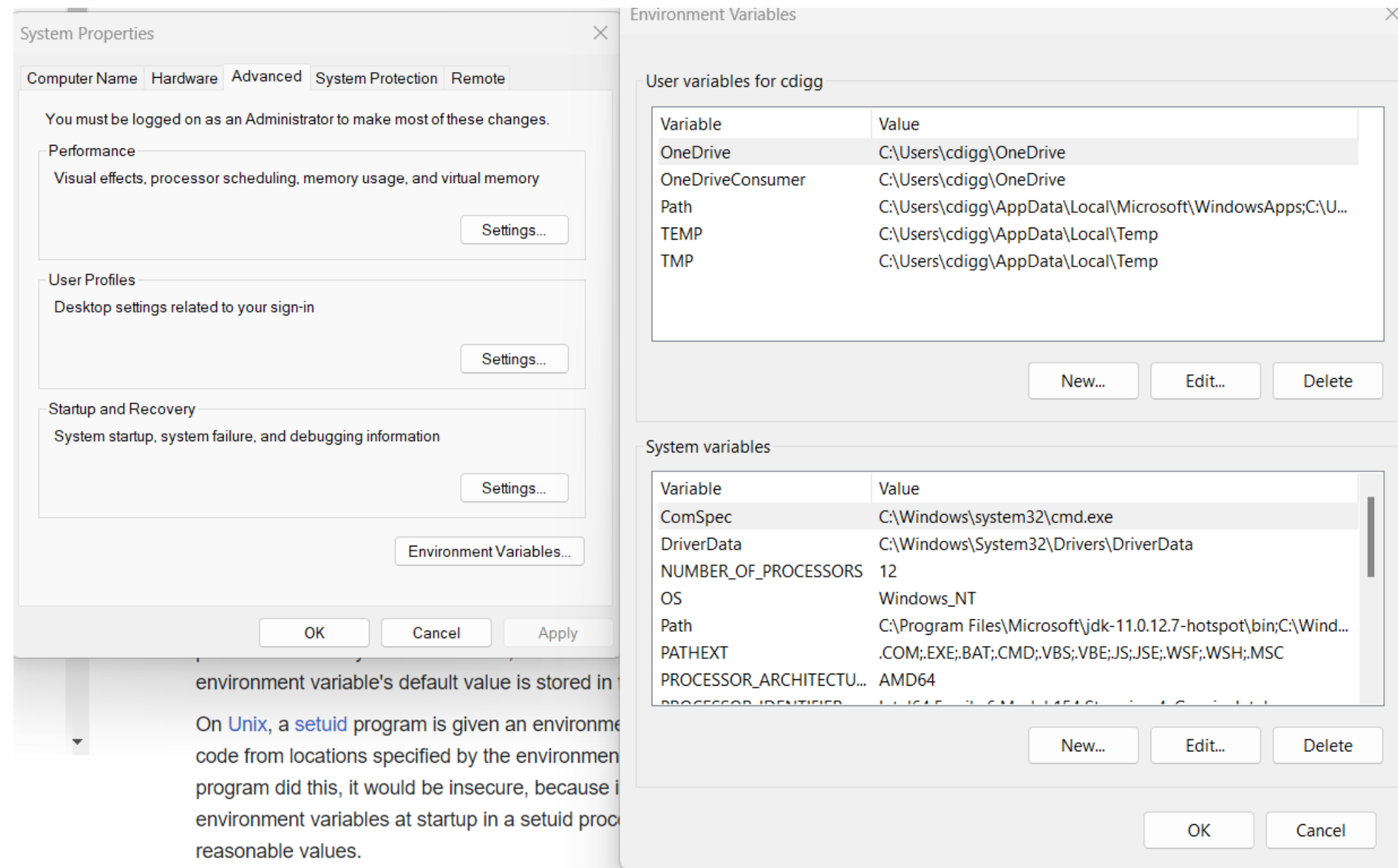


`command | command` – redirects the output of a command to the input of another



More commands: <https://ss64.com/nt/syntax-redirectation.html>

Environment Variable Editor in Windows



Environment Variables

A set of named string values managed by operating system

Each running process has access to its own copy

Process may modify its own values during execution

Used to parameterize behavior without using command-line

You can see them by writing “**set**” from command line

```
C:\Users\cdigg\git\cs321>echo tmp  
tmp
```

```
C:\Users\cdigg\git\cs321>echo %tmp%  
C:\Users\cdigg\AppData\Local\Temp
```

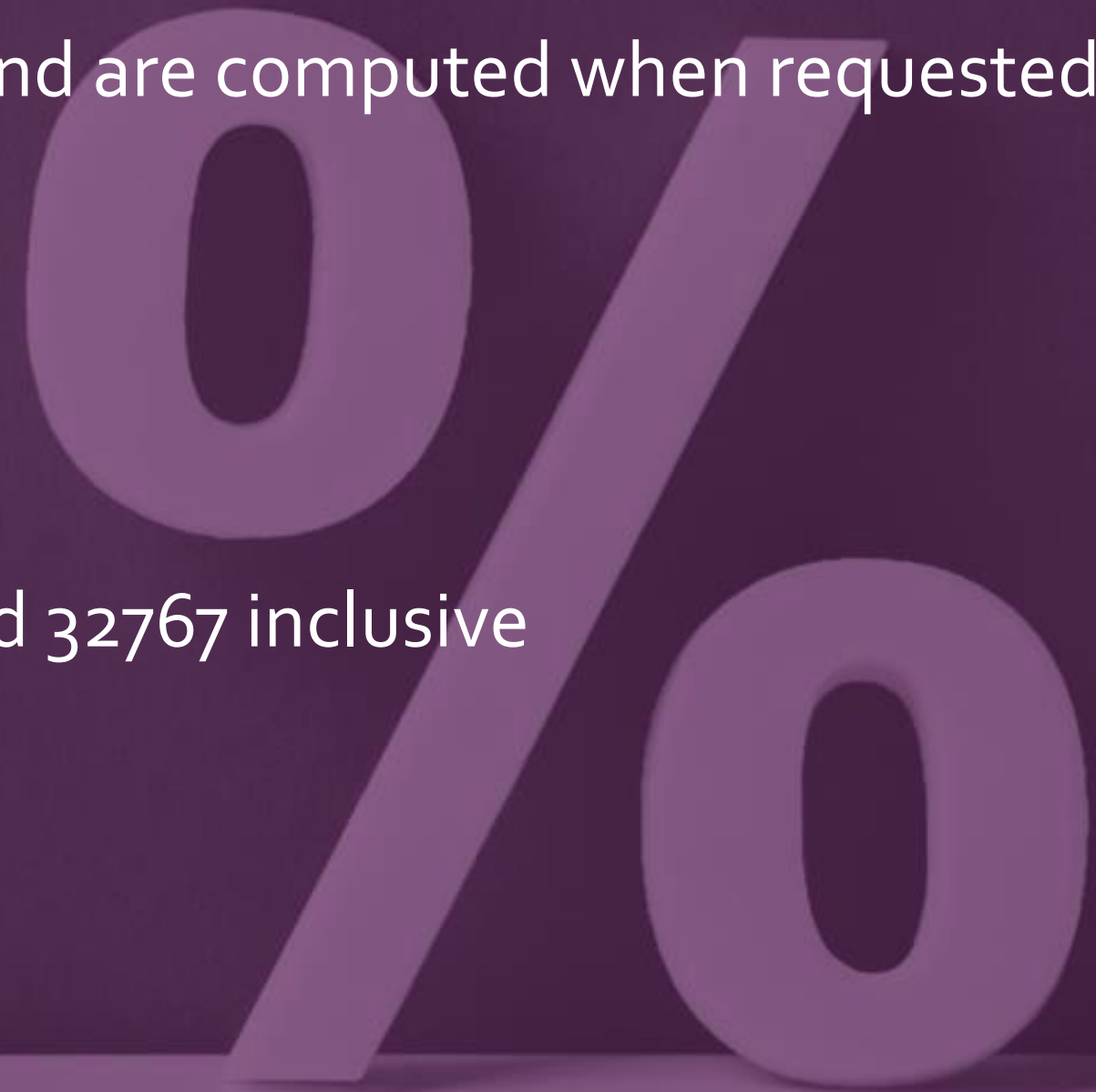
```
C:\Users\cdigg\git\cs321>
```

ENVIRONMENT VARIABLE EXPANSION

On the command-line environment variables are expanded if delimited with %

Pseudo-Environment Variables

- Pseudo-environment variables cannot be set, and are computed when requested
- %DATE% - Current date
- %TIME% - Current time
- %CD% - Current directory
- %RANDOM% - Random variable between 0 and 32767 inclusive



Aborting Process or Ending Input

Ctrl+C – terminates a process

Ctrl-Break – also terminates a process

Ctrl+Z – signals end of input
(Console.WriteLine() returns null)

C# Launching a Process and Redirection

- [Launching a process](#)
- [Redirect standard-input](#) of process
- [Redirect standard-output](#) of process
- [Redirect standard-output](#) of current program