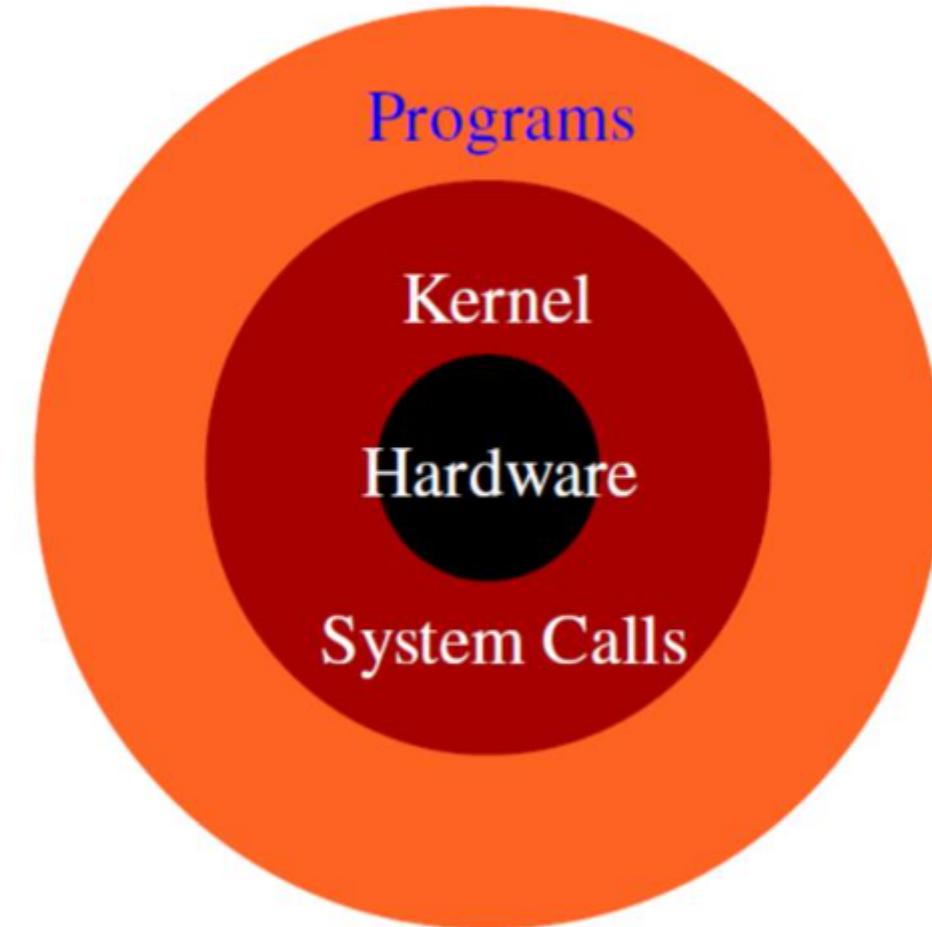


COMP 206 – Intro to Software Systems

Final Lecture – Nov 30th, 2018
Summary, Concluding ideas

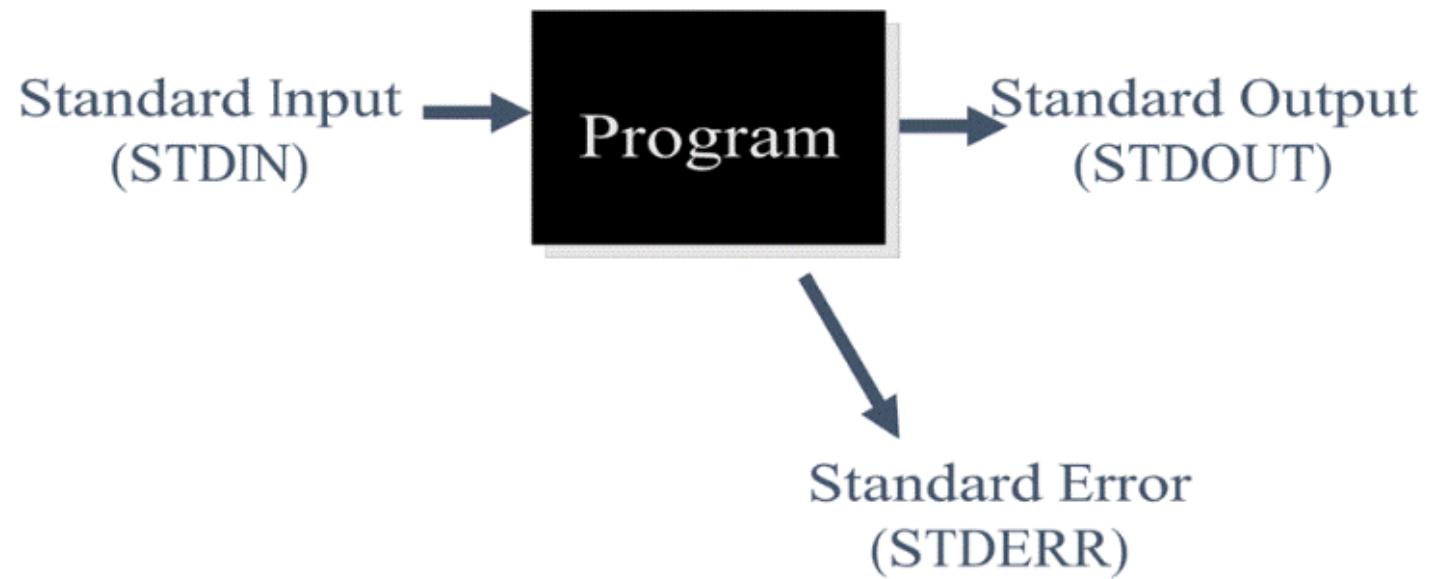
Course Summary (1)

- Your code requires help from many layers to get work done:
 - How are operating systems structured
 - How does the OS help us compile, execute, and run code. The fact that **THIS REALLY MATTERS!**
- Multiple ways to interact with the Operating System:
 - Shell and built-in programs
 - Our own code through C libraries
 - Our own code making system calls



Course Summary (2)

- To do interesting work, our programs need to take input and produce output
- Method 1 was standard input and output. Know how these work by default, and how to change them
- Method 2 was file IO. Text and binary modes important for different cases.



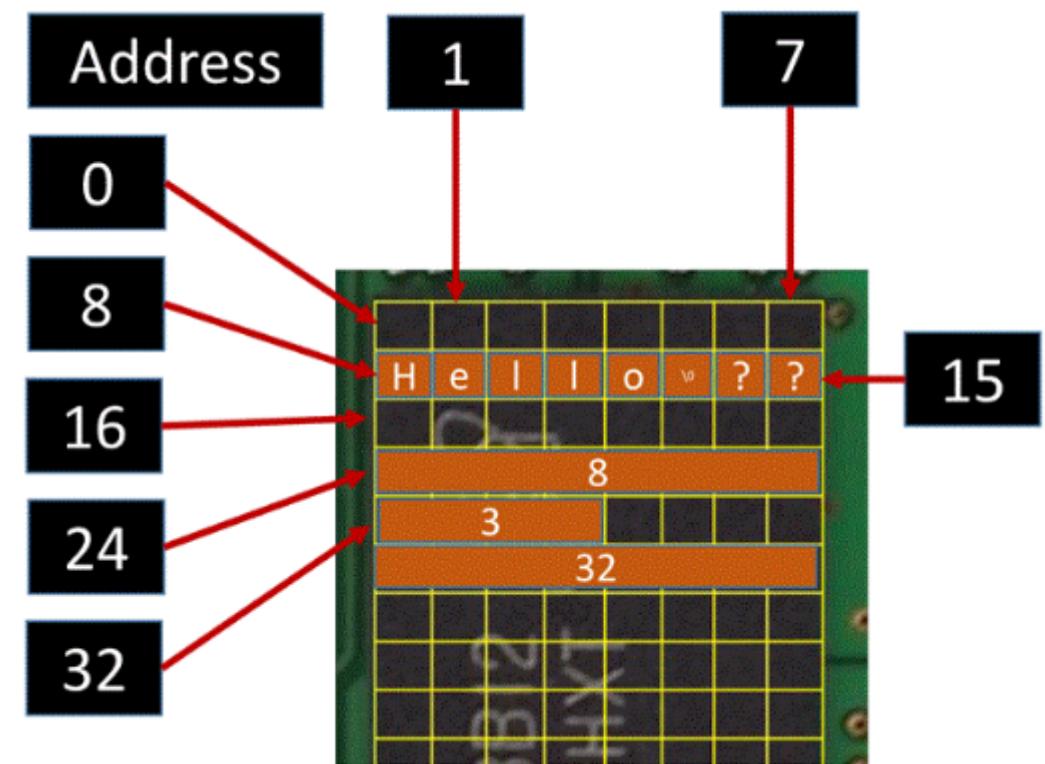
Course Summary (3)

- Data is essential to form useful systems. This data has natural types, but in C we can "mis-use" these.
- For example, packing many "colors" of an image into a 32-bit int
- Knowing these tricks and tools let us handle new data that arises (new user requests, new sensors)

Description	Type	Bits	Range
Integer	short	16	+/-32 thousand
	int	32	+/-2.1 billion
	long	64	+/- 9.2×10^{18}
Floating point	float	32	+/- 10^{38}
	double	64	+/- 10^{308}
	long double	128	+/- 10^{4932}
Character	char	8	-127 to 128
	unsigned char		0 to 255
Pointer	char* int* (etc)	64	0 to 1.8×10^{19}

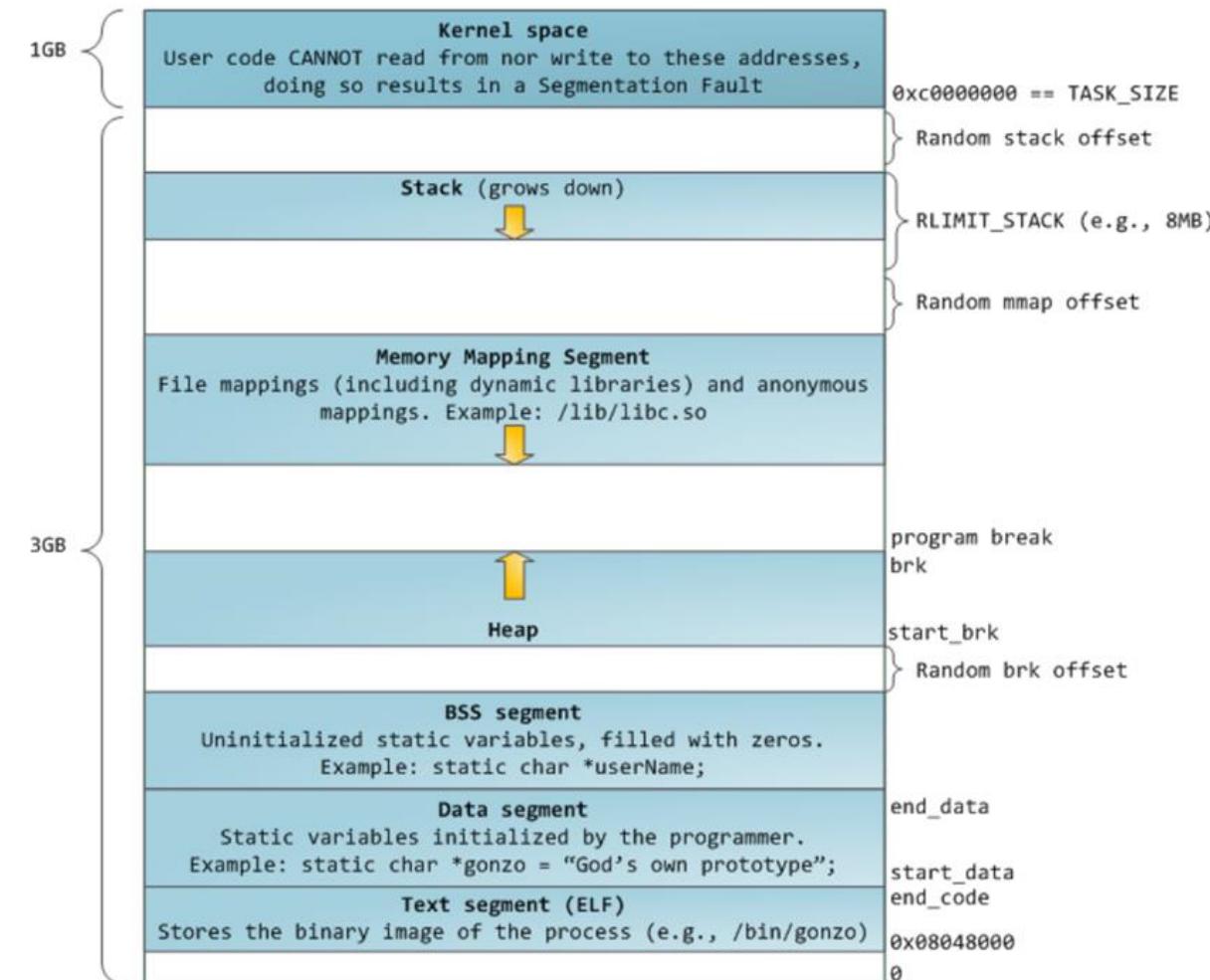
Course Summary (4)

- C strings are character arrays
- C arrays and pointers are nearly equivalent:
 - Important to know the exceptions!
- Addresses to memory and sizes of variables become really important tools
- In C, pointers are used to deal directly with data (e.g., `write()`) and to make code re-usable (function pointers)



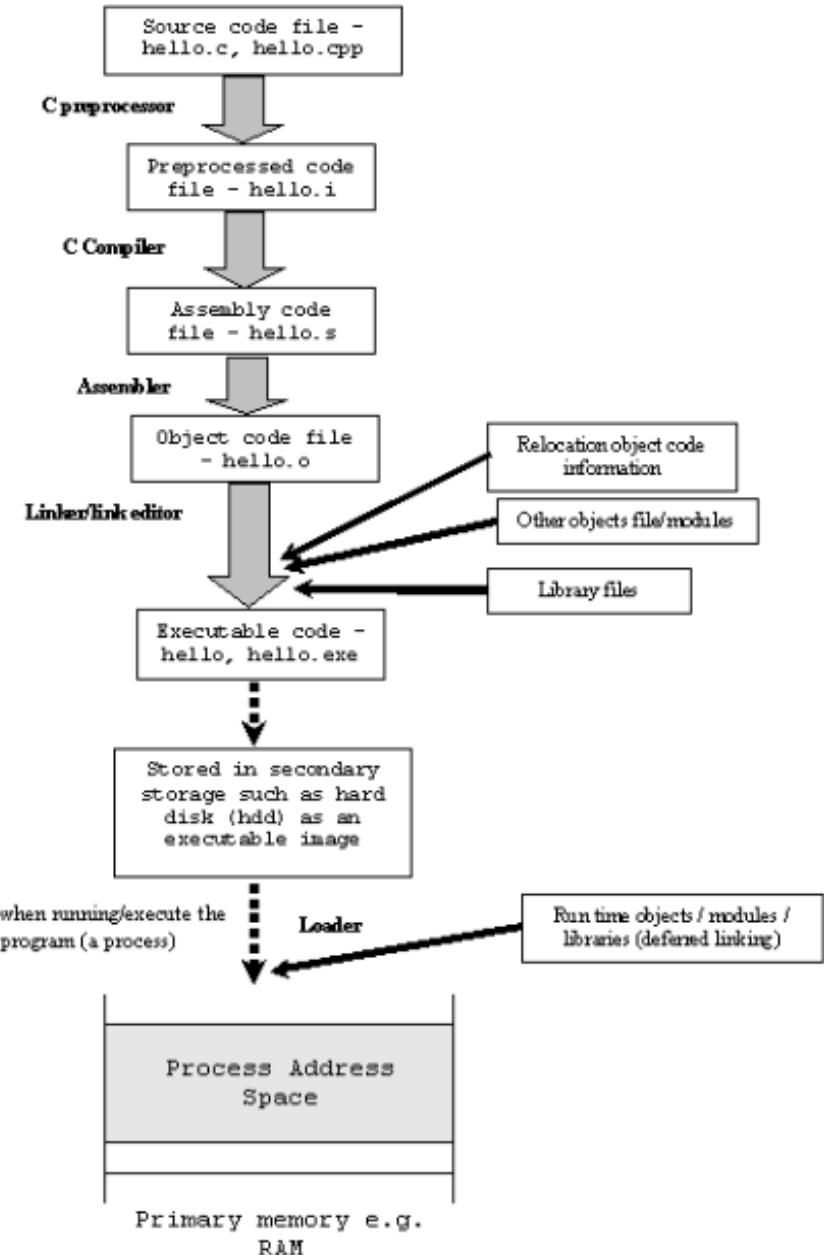
Course Summary (5)

- The structure of a running process in memory has several impacts:
 - Stack memory is temporary
 - Heap memory has programmer-controlled lifetime
- Every process can access the kernel through system calls
- Compiled code is included
- Knowing this allows plug-ins (Adblock), hacking (buffer overflow), multi-process



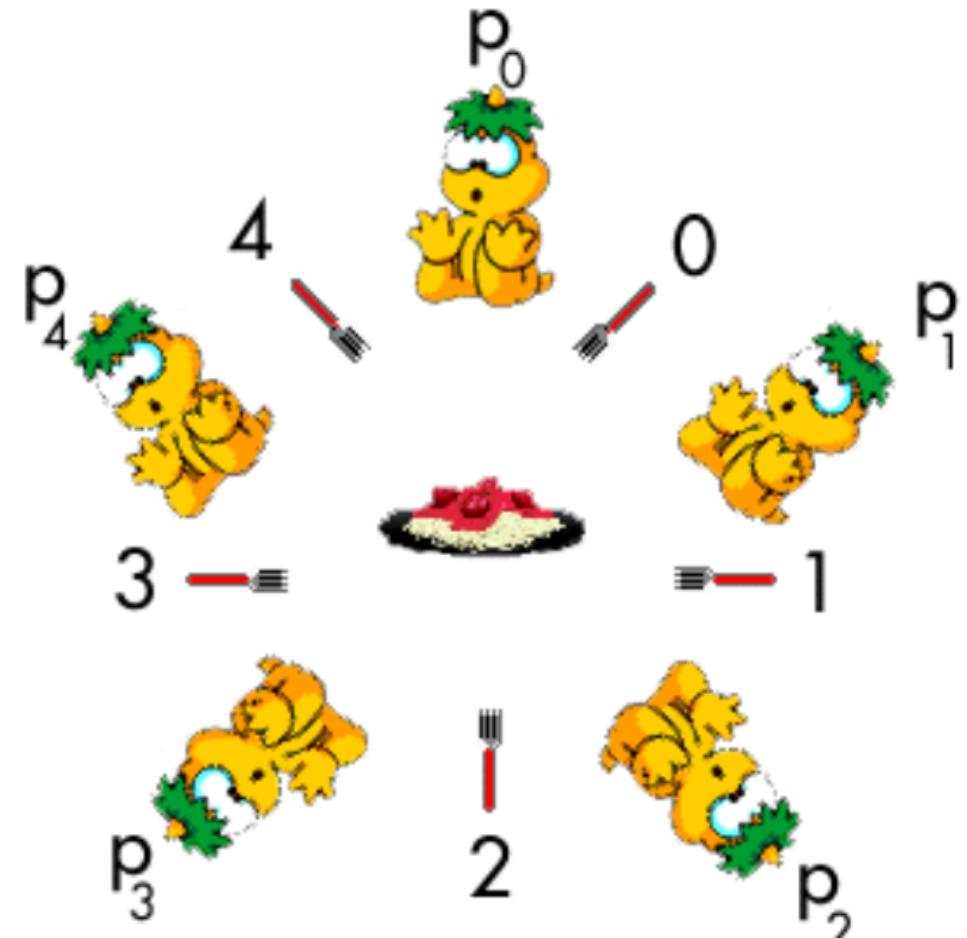
Course Summary (6)

- Compiling in C is not a single step. Knowing about the different phases gives us more opportunity to influence the final program:
 - Pre-processor debugging
 - Using shared libraries to customize code (or hack it)
- The process is usually automated with Makefiles (and even more high-level tools like Cmake, we did not cover this year)



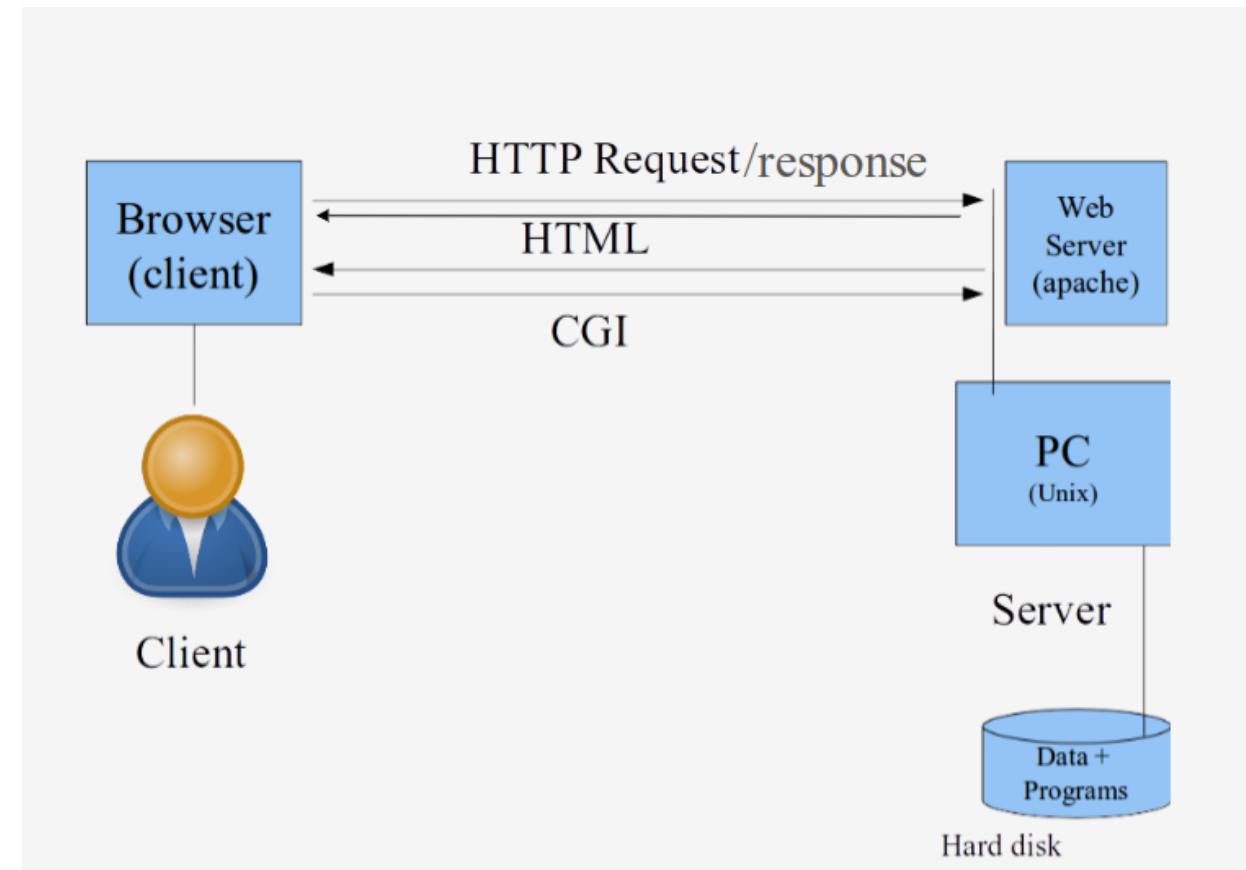
Course Summary (7)

- Software systems are almost always distributed to include many processes, across many machines
- Dining Philosophers was one illustration of how this can go wrong
 - Semaphores are a proposed fix for this proposed by Dijkstra
- Two generals problem is about errorful comms and is impossible to solve perfectly:
 - Today, block-chain solves the Byzantine Generals problem, a variant about consensus in presence of traitors



Course Summary (8)

- The internet forms a world-wide software system:
 - IP includes addressing and builds networks
 - TCP includes ports and congestion control for fair usage by all
 - Each level wraps the previous in packets with headers (similar to BMP)
- The web sits on top of these, with HTTP request/responses defining the web client/server protocol



SOCS Courses Unlocked by 206

- COMP 303 Software Design
 - Principles, mechanisms, techniques, and tools for object-oriented software design and its implementation, including encapsulation, design patterns, and unit testing.
- COMP 307 Principles of Web Development
 - The course discusses the major principles, algorithms, languages and technologies that underlie web development. Students receive practical hands-on experience through a project.
- COMP 310 Operating Systems:
 - Control and scheduling of large information processing systems. Operating system software - resource allocation, dispatching, processors, access methods, job control languages, main storage management. Batch processing, multiprogramming, multiprocessing, time sharing.
- COMP 321 Programming Challenges:
 - Development of programming skills on tricky challenges, games and puzzles by means of programming competitions.
- COMP 322 Intro to C++:
 - Basics and advanced features of the C++ language. Syntax, memory management, class structure, method and operator overloading, multiple inheritance, access control, stream I/O, templates, exception handling.

SOCS Courses Unlocked by 206

- COMP 361 Software Design Project
 - Software development process in practice: requirement elicitation and analysis, software design, implementation, integration, test planning, and maintenance. Application of the core concepts and techniques through the realization of a large software system.
- COMP 417 Introduction Robotics and Intelligent Systems
 - This course considers issues relevant to the design of robotic and of intelligent systems. How can robots move and interact. Robotic hardware systems. Kinematics and inverse kinematics. Sensors, sensor data interpretation and sensor fusion. Path planning. Configuration spaces. Position estimation. Intelligent systems. Spatial mapping. Multi-agent systems. Applications.
- COMP 421 Database Systems
 - Database Design: conceptual design of databases (e.g., entity-relationship model), relational data model, functional dependencies. Database Manipulation: relational algebra, SQL, database application programming, triggers, access control. Database Implementation: transactions, concurrency control, recovery, query execution and query optimization.
- COMP 424 Artificial Intelligence
 - Introduction to search methods. Knowledge representation using logic and probability. Planning and decision making under uncertainty. Introduction to machine learning.

A few final points to ponder

- The C compiler is one of humanity's great achievements:
 - It starts the chain of re-usable code. More C programs built on top: the compilers and interpreters of other languages, the web
 - Modern C compilers are written in C. This makes C self-replicating...
 - How was the very first C compiler written, and what if a bug had been inserted?
- You are part of the open-source community.
 - It's OK to move any 206 material to your own account on github.com if you wish.
 - I hope you keep using these tools. It's good for you and for the world!

Where are software systems outside of McGill?

