C++

Data Encapsulation

Polymorphism

Object oriented paradigm - Class/Objects

Static typing - basic datatypes, const

Virtual functions

Pointers

References

Templates

Python

Discussed dynamic typing

Interpreted vs. compiled languages

Many ideas are borrowed from other languages, e.g., object-orientation from C++/Simula

Better scripting, string manipulation, library support

Dynamic typing makes it easier to code

Graph

C++ code for a graph data structure

Breadth-First Search

Systems

-------

Processes, forks, pipes

What is a process

Memory layout of a process

forks, pipes, signals and their uses

Control-hijack Attacks

Basic buffer overflow, integer overflow, and string format attacks

Return-oriented programming

Basic protection mechanisms, including canaries, NX bit, address space layout randomization (ASLR)

Regular Expressions

-------------------

Definition

Expressing languages via regexes

Given a language description, provide a regex

Given a regex, provide a language description

Post midterm

++++++++++++

Concurrency (study code examples given in class)

------------

Multi-programming

Multi-threading

Difference between processes and threads

Concurrency vs. Parallelism

Race condition in multi-threaded programs

Mutexes

Semaphores

Producer-consumer with Semaphores

Propositional Logic

-------------------

What is logic?

Recursive definition of syntax and semantics of prop. Logic

Normal forms - NNF, CNF, DNF

The Boolean SAT problem

Why not DNF for satisfiability?

Why CNF for satisfiability?

Duality between satisfiability and validity problems

SAT solvers

-----------

The definition of the SAT problem

Basics of DPLL SAT solver

# Regular Expressions

Ф：空集 没有任何元素 L（Ф）=Ф

λ：空字符串 表达式中的元素就是空 L（λ）={λ}

r= xxx ：正规式Regex L={xxx}：正规式的描述Language

L = {w | w contains no sub-string 01}. 1\*0\*

L = {w | w contains no sub-string 00} （1+01）\*（0+lamda）

# Overflow

Buffer和

# Concurrency

## Concurrency

Concept: how do we optimally use shared resources (e.g., one CPU) among processes or threads.

Key concept is interleaving via **context-switching**: To accomplish this the OS scheduler context-switches process P1 out (i.e., save its registers) and context-switches P2 in (i.e., writes its saved register values into the CPU registers), whenever P1 starts big I/O access

把当前正在执行的任务状态保存下来，然后加载下一个任务状态并执行

## Parallelism

Concept: how to use multiple resources (e.g., multi-processors) optimally (minimize time).

Key concepts is to identify and minimize dependencies between tasks (threads or processes) so that they can be run as independently as possible.

## Multiprogramming

Concurrent execution of multiple tasks (e.g., processes). Each task runs as if it was the only task running on the CPU.

Benefit: When one task needs to wait for I/O, the processor can switch to another task.

## Multithreading

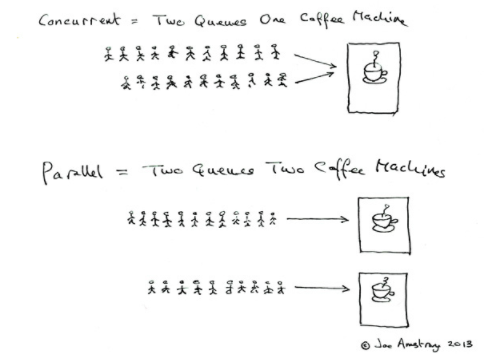
## Concurrency and Parallelism

Concurrency is when two or more tasks can run in overlapping time periods, not necessarily in parallel. In this context, two threads or processes may or may not run simultaneously on a single-core machine. The core idea here is to share resources when one process is waiting on I/O or memory access etc.

Parallelism is when tasks/programs run at the same time on a multicore processor or a network of workstations etc. (best modern examples include GPUs, your laptop microprocessors,…). Dominant models include shared memory and MPI.

并发和并行区别：并发（concurrency）：把任务在不同的时间点交给处理器进行处理。在同一时间点，任务并不会同时运行。Multiple programming是concurrency的一种情况

并行（parallelism）：把每一个任务分配给每一个处理器独立完成。在同一时间点，任务一定是同时运行。



An assignment A is suitable/complete for a formula F if A assigns a truth value to every atomic proposition of F

An assignment A is a model for F, written A⊧F, iff A is suitable for F A’(F) = 1, i.e., F evaluates to true (or holds) under A

A formula F is satisfiable iff there exists at least one model for F. Otherwise, we say F is unsatisfiable (or contradictory)

A formula F is valid(or a tautology), written ⊧F, iff every suitable assignment for F is a model for F

A formula F is valid if and only if ¬F is unsatisfiable, A formula F is not valid (invalid) if and only if ¬F is satisfiable

真值表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P | Q | P∧Q | P∨Q | P→Q | P⇿Q |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

手机屏幕截图

描述已自动生成

文本, 信件

描述已自动生成

文本

描述已自动生成