

Lecture 1 Inf2C, Computer Systems: Course overview & the big picture

Stratis Viglas

School of Informatics
University of Edinburgh



Inf2C - In general

- 20 credit course covering software engineering and systems
- Two threads - independently taught
 - Perdita Stevens teaches SE
 - Marcelo Cintra and Stratis Viglas teach systems
- Threads alternate week by week
- Independent practicals, tutorials, **common exam**
- All material will be on course web-page:
www.inf.ed.ac.uk/teaching/courses/inf2c
 - Especially check the “schedule” page



Inf2C - practicalities

- Lectures:
 - Tue, Thu, Fri, AT LT3 @ 3pm
- Tutorials
 - Groups just been assigned. Check your email
 - Start at week 3
- Exam - 75%
 - In December; exact date not available yet.
 - Common section with MCQs
 - 2 other sections (one SE, one CS) with 2 questions each
- Coursework - 25%, min cw mark: 25/100
 - Four assignments
 - Check schedule web page for deadlines



Inf2C - student representatives

- We need ~ 2 per Inf2 course
 - Apparently, these have already been assigned by Perdita
- Duties
 - Point of contact for suggestions, complaints of general concern
 - Attend a staff-student liaison meeting to discuss the above



Late coursework

- New School-wide consistent policy:
- **Normally, you will not be allowed to submit coursework late**
- If you have a good reason to submit late, email the Inf2 course organiser, copying to the lecturer concerned as soon as possible
 - Keep your DoS updated too!
 - Only in exceptional circumstances (*e.g.*, illness that stopped you getting to email), would an extension be granted after a deadline has passed
- See course guide for details
 - www.inf.ed.ac.uk/teaching/years/ug2/inf2Guide.html



Good reason

Something that, in the judgement of the member of staff responsible, would prevent a competent, well-organised, conscientious student from being able to submit on time. E.g.:

- Significant illness
- Serious personal problems
- Interviews/selection procedures, in some circumstances

Non-examples:

- Difficult cluster of deadlines

Last-minute computer problems, (your own) back up failure, ...



Inf2C – systems thread

- This is the systems thread of Inf2C
 - 14 lectures
- Hardware
 - Data representation and operations
 - Processor organisation & design of simple circuits
 - Exceptions and interrupts
 - The memory sub-system
 - Input/Output
- Software:
 - Low-level programming
 - Operating systems basics
 - Introduction to C programming (2 lectures)



Inf2C Systems – Learning outcomes

- Demonstrate an understanding of binary representation and basic operations on binary data.
- Demonstrate an understanding of key concepts in computer architecture, including: exceptions, interrupts, virtual memory, processes and pipelined execution.
- Sketch the design of a simple processor and explain how it operates.
- Demonstrate knowledge of I/O devices and the means by which they interface to a processor and its memory system.
- Demonstrate an understanding of the design and operation of important combinational and sequential components within a processor, such as adders, registers, and state machines.
- Demonstrate understanding of an execution pipeline, based on the MIPS architecture.



Inf2C Systems – activities

- Coursework – 2 assignments for systems thread
 1. MIPS (Microprocessor without Interlocked Pipeline Stages) assembly programming
 2. Implement the control unit of a simplified MIPS processor in system-C
- Drop-in labs:
 - Demonstrators available to provide help
 - Lab scripts to try out; not assessed
 - Inf2A students: **check for clashes with tutorials and labs**
- Tutorials:
 - Half of the tutorials are on systems
- Notes are provided, but you **must** read the book too
- Newsgroup: `eduni.inf.course.inf2c`



Inf2C Systems – books

- Patterson & Hennessy: Computer Organization and Design, Morgan Kaufmann
 - Third edition available
 - Library has several 2/e copies, still OK

Worth buying if this is the only course on computer architecture/hardware you will ever take
- Silberschatz, Galvin, Gagne: Operating Systems Concepts, Willey 7/e (library copies 5/e)
 - Only a few sections needed for this course



Evolution of computers

- Early computers had their programs set up by plugging cables and setting switches
- **John von Neumann** first proposed to store the program in the computer's memory
- All computers since then (~1945) are stored-program machines

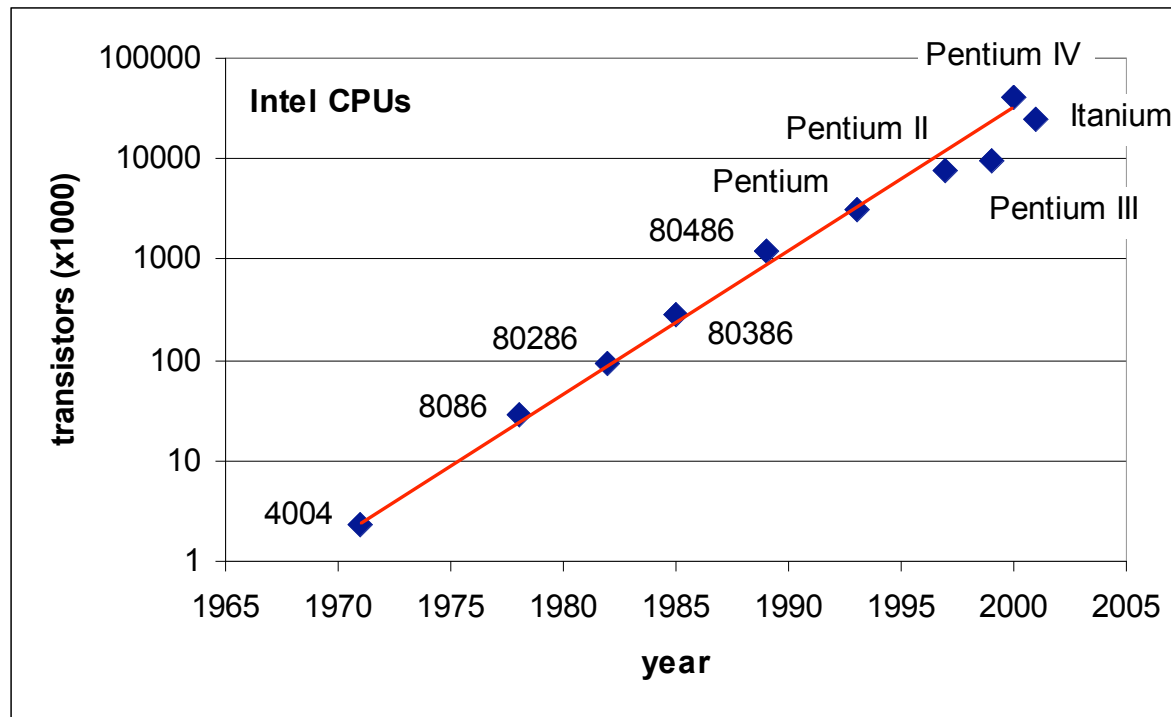


Evolution of computers

- What has changed is the number of transistors (switches) and their speed
- Implementation technology progressed from tubes to discrete bipolar transistors, MOS transistors, and Integrated Circuits (chips)
- At the same time, the cost per transistor has been dropping



Moore's Law



- Transistor counts roughly double every 18 to 24 months
 - Intel 4004 (year 1971): 2300 transistors
 - Intel Pentium IV (year 2000): 42,000,000 transistors

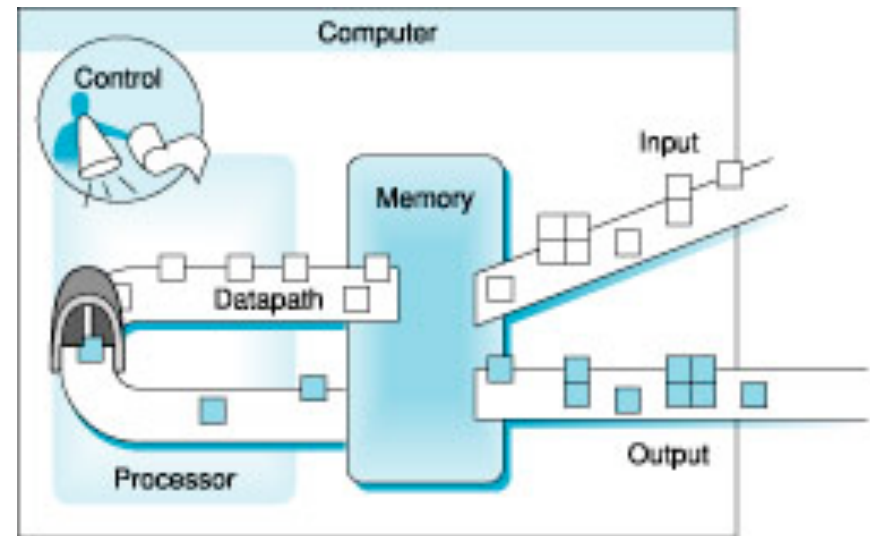
10,000x increase in 30 years!



Computer components

- Data path
 - Performs actual operations on data
- Control path
 - Fetches instructions from program in memory
 - Requests operations on data from data path also in order
- Memory
 - Stores data and instructions
- Input/Output
 - Interfaces with other devices for getting/giving data

} Processor



Types of computer systems

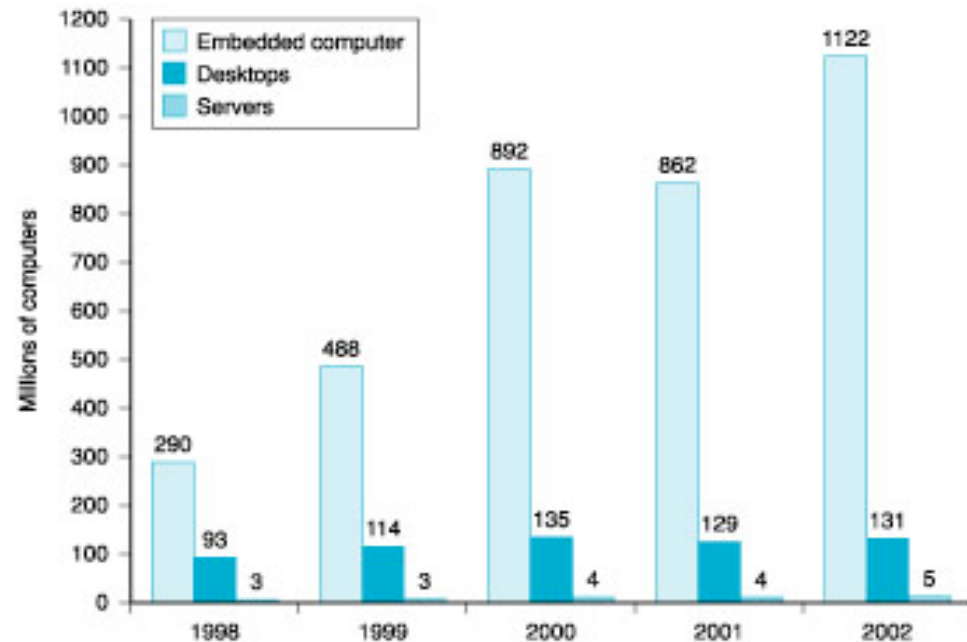
- Servers
 - Fast processor(s), fast I/O
 - Used for either few large tasks (engineering apps), or many small tasks (web server)
 - Multi-user, multi-program
- Desktops
 - The common PC
 - Balance cost, processing power
 - Single/multi-user, multi-program



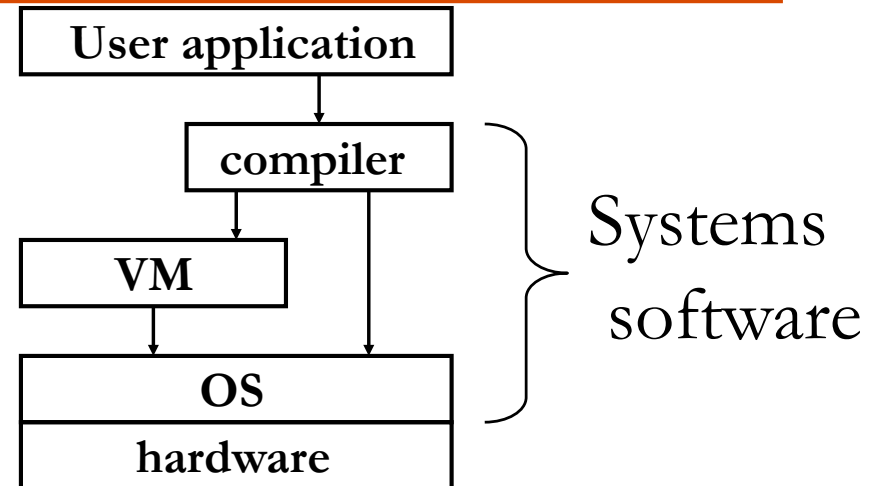
Types of computer systems - 2

- Embedded:
 - Computing not main purpose of the device
 - Low-cost, low-power (for portable devices)
 - Single user, usually single program, not user programmable

- Which is the largest category?
Embedded computers



Modern computer system



- Operating System (OS)
 - Mediates access to hardware resources (CPU, Memory, I/O)
 - Schedules applications
- Compiler
 - Translates **High Level Language (HLL)** into **machine language** or **byte code**
- Virtual Machine (VM)
 - Interprets and “executes” byte code

