Operating Systems (SMH) (Part IA)
Paper 1 Question 11 2004 (20 Marks)

Note: overall, this question is mainly from the first section of the course. However the final part links this with the core "operating systems" lectures.

Von Neumann

Key innovation: sharing of instructions and data within a single "memory" (i.e. the notion of a stored program). Prior to this, "programs" were effectively hard-coded.

Data in main memory

Note: this is from a series of a slides in the first section of the course in which the students learn the way in which various forms of data might be stored in the memory of a simple computer.

Unsigned integer: just binary number interpreted as an integer between 0 and $2^w - 1$, where w is the word size (in bits).

Signed integer: 2's complement – viz. invert (complement) all bits and add 1 to perform unary minus. I expect a diagram or indication in this case about what is different from the unsigned case.

Floating point number: as a exponent and mantissa; the exponent in (signed) bias-2^e format and the mantissa using "binary point" (obvious extension of binary to interpret bits $b_{-1}b_{-2}b_{-3}\ldots$ as $b_{-1} \times 2_{-1} + b_{-2} \times 2_{-2} + b_{-3} \times 2_{-3} + \ldots$). *Note: don't require discussion of normalization/denorms or IEEE stuff here.*

Instruction: as a structured set of bit fields in which various parts denote the "instruction" and various others the operand(s). Miscellaneous other flags may also be present (e.g. write-back result?, set flags?, etc). *Note: anything plausible will work here – they've seen the ARM as an example in the lectures.*

OS and data types

No; it neither knows nor cares. Just treats registers as 'bits'.

Context switch

During a context switch the operating system is entered (either explicitly via e.g. a yield or system call, or implicitly via a timer interrupt) and the *context* the currently executing process is stored. This means the register state among other things. Once it is decided in whatever fashion which process to execute next, *its* context is restored from where it was previously saved. Key thing here for this question is to note that the contents of registers can be explicitly stored and loaded from memory.