Short Notes Chapter 16 (Lecture 44):

Calling conventions of the language:

To interface an assembly routine with a high-level language program means to be able to call functions back and forth. And to be able to do so requires knowledge of certain behavior of the HLL when calling functions. This behavior of calling functions is called the calling conventions of the language.

Two prevalent calling conventions are:

- the C calling convention &
- the Pascal calling convention

What is the naming convention?

C prepends an underscore to every function or variable name while Pascal translates the name to all uppercase. C++ has a weird name mangling scheme that is compiler dependent. To avoid it C++ can be forced to use C style naming with extern "C" directive.

How are parameters passed to the routine?

In C parameters are pushed in reverse order with the rightmost being pushed first. While in Pascal they are pushed in proper order with the leftmost being pushed first.

Which registers must be preserved?

Both standards preserve EBX, ESI, EDI, EBP, ESP, DS, ES, and SS.

Which registers are used as scratch?

Both standards do not preserve or guarantee the value of EAX, ECX, EDX, FS, GS, EFLAGS, and any other registers.

Which register holds the return value?

Both C and Pascal return up to 32-bit large values in EAX and up to 64-bit large values in EDX:EAX.

Who is responsible for removing the parameters?

In C the caller removes the parameter while in Pascal the callee removes them. The C scheme has reasons pertaining to its provision for variable number of arguments.

👉 Also Recommended: CS401 SHORT NOTES CHAPTER 1-7 Midterm