Quando um programa escrito em hll é executado, o que define o estado dessa execução?

* código (ponto de execução) -> REG
* variáveis

(estáticas -permanentes; [globais] -> memória [DATA]

automáticas - temporárias) [locais] -> memória [STACK]

* histórico de invocação de sub-rotinas -> STACK
* regs guardam os endereços relevantes na memória. (no text)
* text -> codigo
* data -> variaveis estáticas
* malloc -> heap
* variaveis auto. , hist. subrotinas

Como esse programa é executado no hw?

1. Tradução (compilação/interpretação) alto nível -> linguagem de máquina.
2. Execução

* Processamento da linguagem de máquina pela CPU
* Interação com os dispositivos E/S

Como a CPU controla o fluxo de execução?

* Registradores -> [memória fica implícita]

Como a CPU sabe quais endereços contêm instruções e quais contêm dados?

* Não sabe.

Como aplicação e SO podem coexistir no sistema?

* Na memória, espaço para aplicação e outra pro so.

Como impedir que uma aplicação modifique indevidamente o so?

* 2+ níveis de privilégio [modos de operação]