Heterogeneidade

Aluno: Cleber Peter.

Disciplina: Sistemas Distribuídos.

Professor(a): Adenaur Correa Yamin.

28/03/20

UNIVERSIDADE CATÓLICA DE PELOTAS

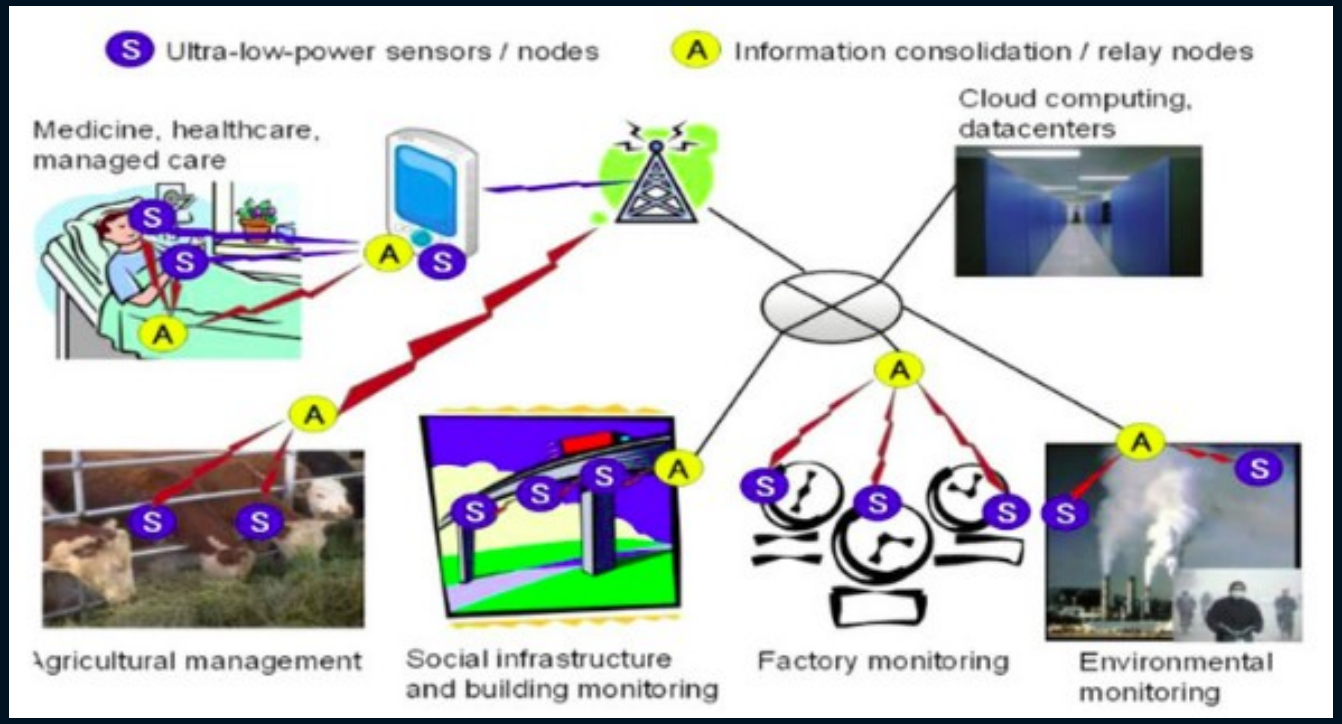
MESTRADO ENGENHARIA ELETRÔNICA E COMPUTAÇÃO

Roteiro da apresentação

- Conceito
- Problematização
- Camadas de abstração
- Conclusão

Conceito

Diferentes sistemas de software e hardware, com requisitos e características distintas, integram-se em uma mesma rede.

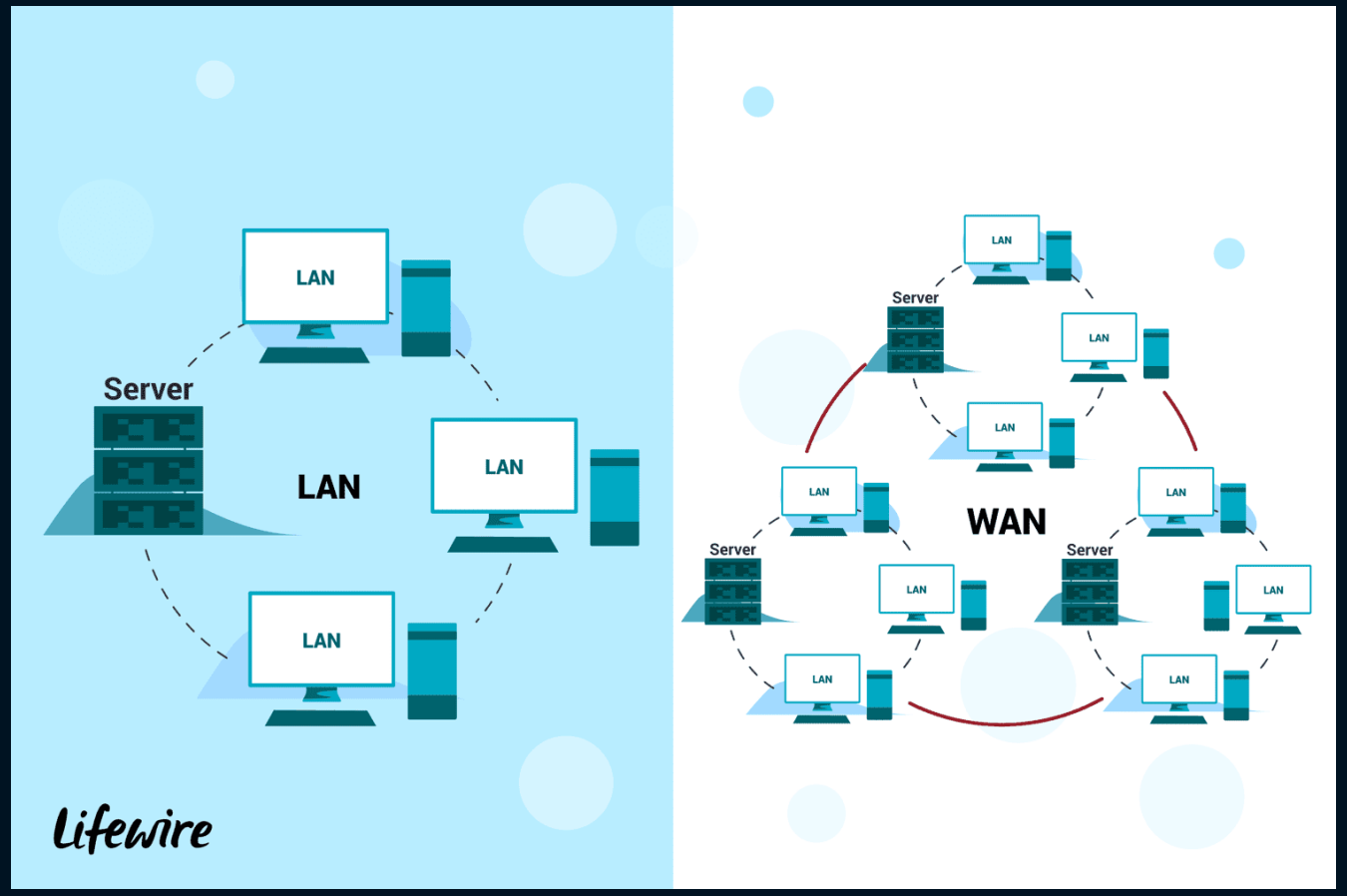


Aplicabilidade

- Redes
- Hardware dos dispositivos
- Sistemas operacionais
- Linguagens de programação
- Implementações distintas

Redes Heterogêneas

Os dispositivos são alocados em redes distintas.

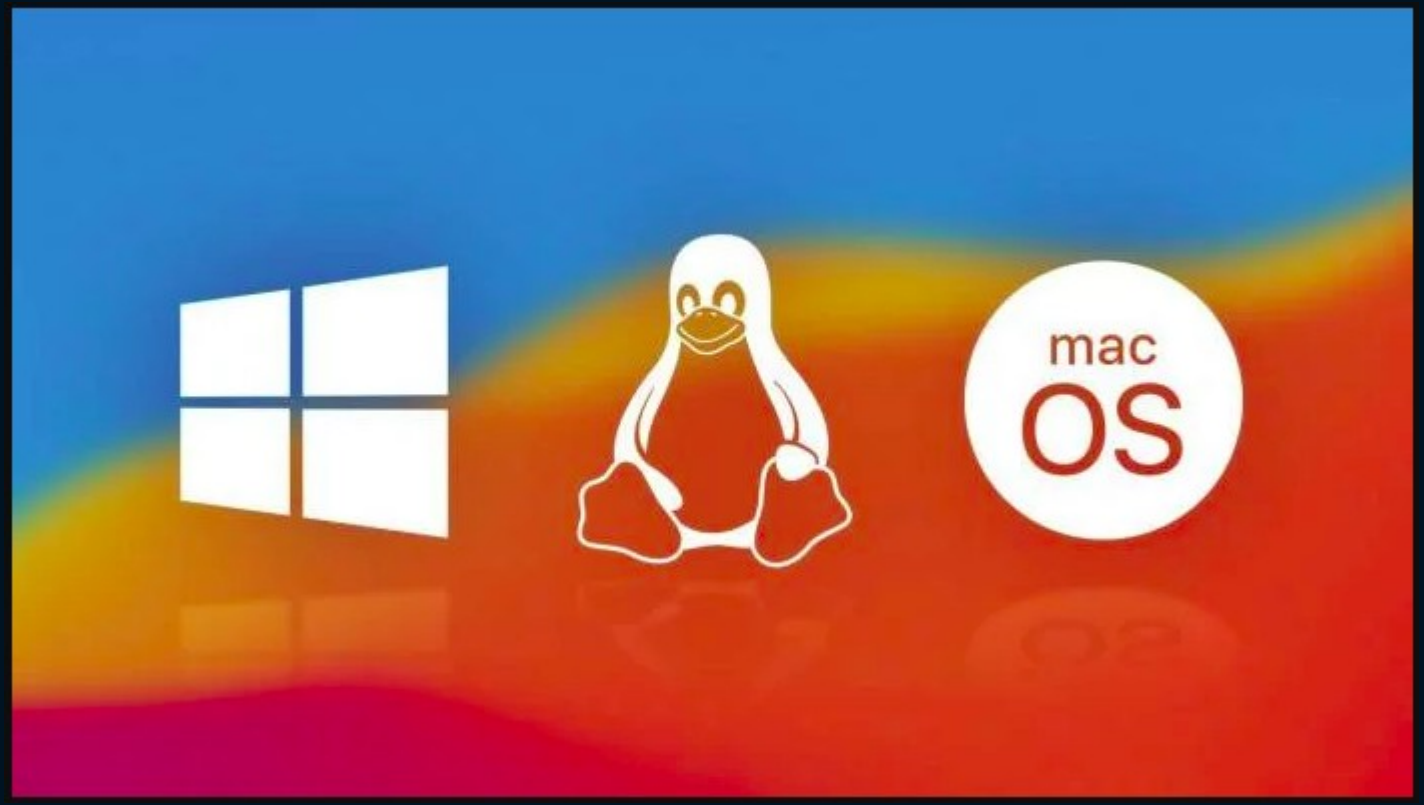


Hardwares Heterogêneos

Os hardwares possuem arquiteturas distintas (x86, x64, ARM, MIPS).

Sistemas Operacionais Heterogêneos

Os dispositivos possuem sistemas operacionais distintos (Linux, Windows, MacOS, Bare Metal, Android, IOS).



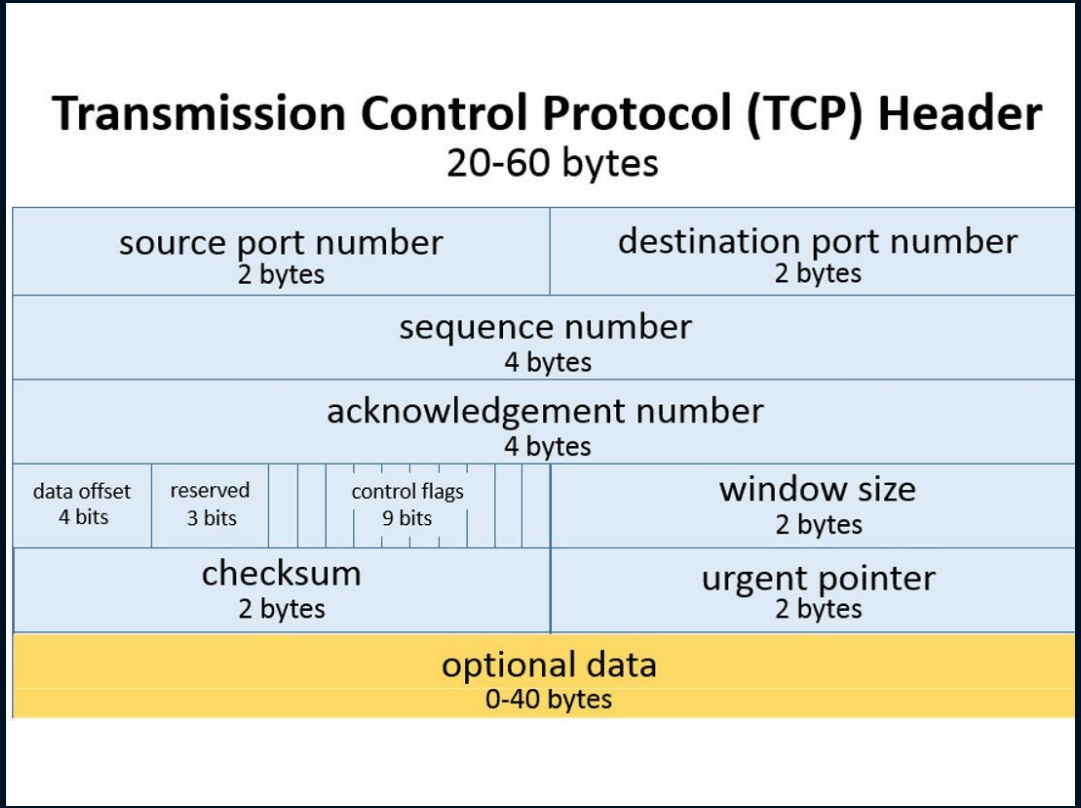
Linguagens de Programação Heterogêneas

Os softwares são programados em linguagens distintas as quais nem sempre possuem a mesma tipagem de dados.

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		99.7
3. C		96.7
4. Assembly		74.1
5. Arduino		69.0
6. Haskell		48.6
7. VHDL		45.4
8. Verilog		41.2
9. D		40.6
10. LabView		32.7
11. Erlang		26.9
12. TCL		21.9
13. Ada		20.9
14. Ladder Logic		11.5
15. Forth		0.0

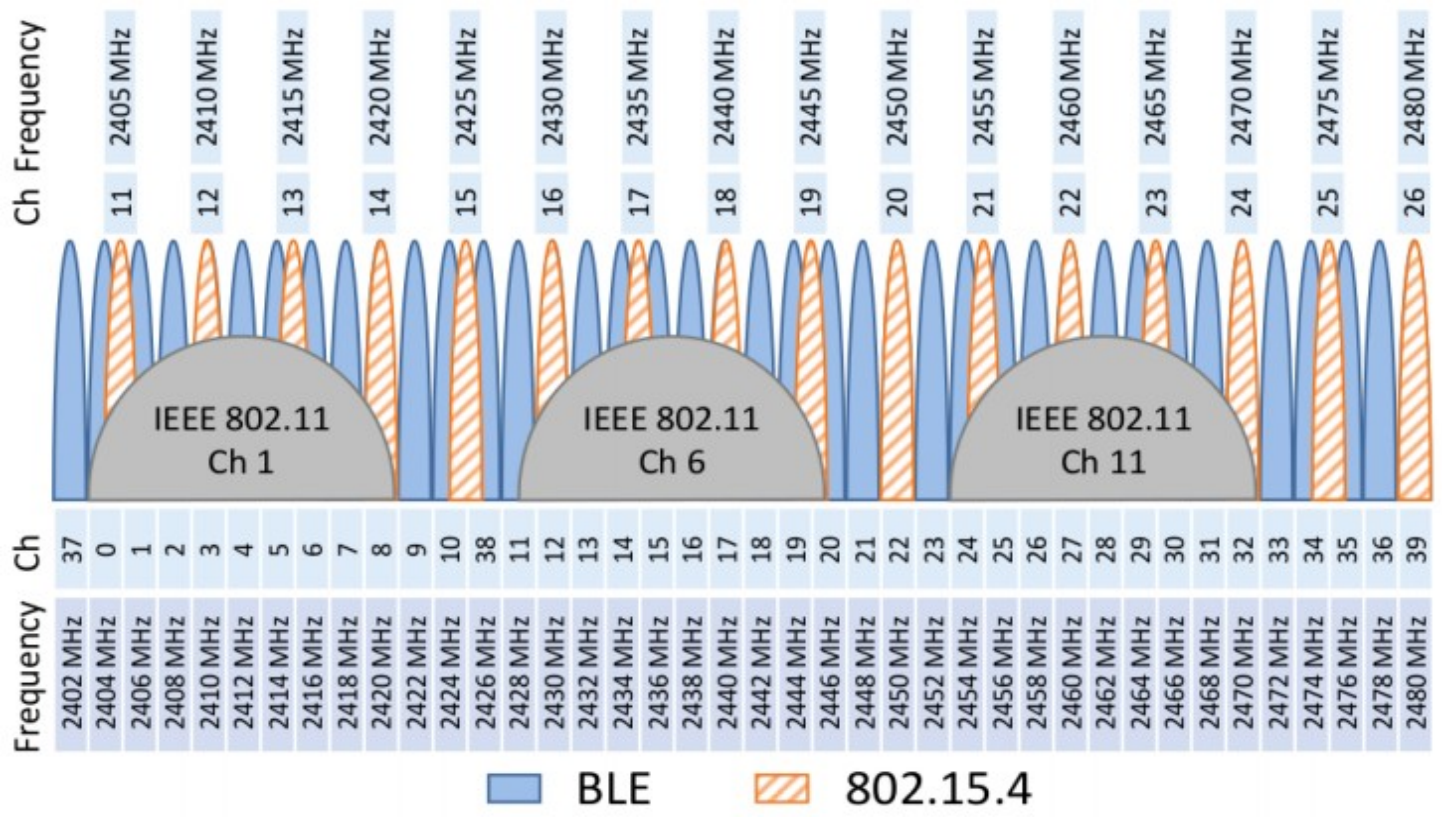
Implementações Heterogêneas

Os programas são escritos por programadores diferentes. Para que haja comunicação entre estes é necessário que padrões/protocolos sejam estabelecidos.



Interferência em Ambientes Heterogêneos

Sobreposição dos canais na frequência de 2.4Ghz.



Abstração – Arquitetura de Hardware

Representação Numérica:

80 = 1010000b - Big Endian
80 = 0000101b – Little Endian

Cliente com CPU intel (x86,x64) → Little Endian

Servidor ARM (sensor de temperatura) → Little Endian

IBM Mainframes → Big Endian

Conexão TCP/IP (Big Endian) na porta 80.

Abstração – Arquitetura de Hardware

Código no Cliente Intel:

tcp.port = 80; // carrega 0000101b. Afinal 80 LE = 5 BE
tcp→connect(); // tentará conectar na porta 5.

glibc htons function:
tcp.port = htons(80); // adapta o 80LE para 80BE
tcp→connect(); // tentará conectar na porta 80.

* A macro htons detecta a arquitetura do cliente e se este já estiver no formato BE ela não efetua a conversão.

htons glibc

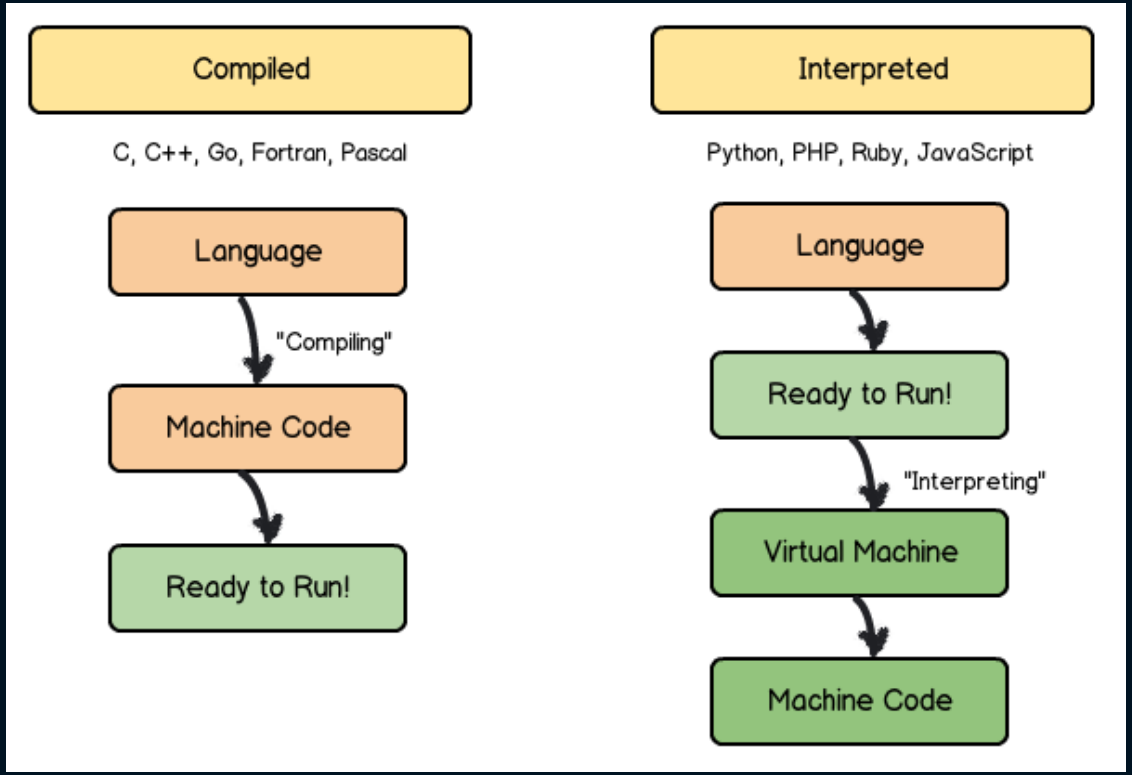
```
#include <netinet/in.h>

#undef        htons
#undef        ntohs

uint16_t
htons (uint16_t x)
{
#if BYTE_ORDER == BIG_ENDIAN
  return x;
#elif BYTE_ORDER == LITTLE_ENDIAN
  return __bswap_16 (x);
#else
# error "What kind of system is this?"
#endif
}
weak_alias (htons, ntohs)
```

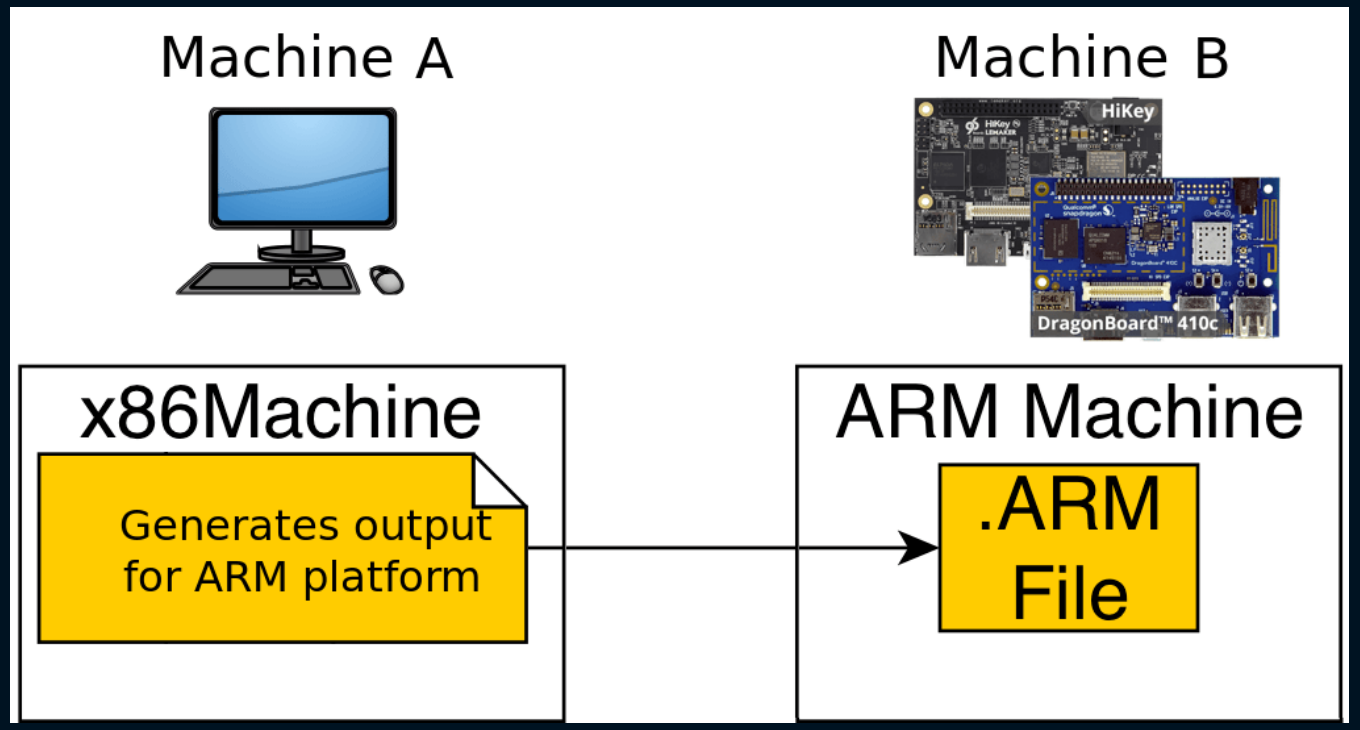
Abstração – Sistemas Operacionais

Uma alternativa para a portabilidade de softwares a distintos sistemas operacionais são as linguagens interpretadas como Java e Python.



Abstração – Sistemas Operacionais

Uma alternativa de melhor desempenho para a portabilidade de software são os frameworks como o Qt, que viabilização a compilação cruzada.



Perguntas?