5-4: Arch intrinsics and inline assembly (Practice)

Artem Pavlov, TII, Abu Dhabi, 07.05.2024

Create new crate

- Create new branch in the repository p54
- Create new library crate p54
- Check that p54 is listed as a member of the workspace in the root Cargo.toml

Implementing AES-128 using AES-NI

- Implement the following AES-128 functions:
  - fn expand_key(key: &[u8; 16]) -> [__m128i; 11]: expands 128-bit key to round keys
  - fn encrypt1(keys: &[__m128i; 11], block: &mut [u8; 16]): encrypts one 128-bit block
  - fn decrypt1(keys: &[__m128i; 11], block: &mut [u8; 16]): decrypts one 128-bit block
  - fn encrypt8(keys: &[__m128i; 11], block: &mut [u8; 128]): encrypts eight 128-bit block
  - fn decrypt8(keys: &[__m128i; 11], block: &mut [u8; 128]): decrypts eight 128-bit block
- Check that AES-NI is available at runtime using is_x86_feature_detected!
- Add tests and benchmarks

Extra task

- Implement CTR mode (with 64-bit counter) for AES-128 using AES-NI and SSE2 intrinsics with the following signature:

```
fn apply_keystream(key: &[u8; 16], data: &mut [u8])
```

Hints

- Intrinsics to use:
  - _mm_loadu_si128, _mm_storeu_si128
  - _mm_shuffle_epi32, _mm_slli_si128, _mm_xor_si128
  - _mm_aesenc_si128, _mm_aesenclast_si128, _mm_aesdec_si128, _mm_aesdeclast_si128, _mm_aeskeygenassist_si128