noalloc, nolock

https://discourse.llvm.org/t/rfc-nolock-and-noalloc-attributes

102

- Similar to `noexcept`

- `[[clang::noalloc]]`

- **`noexcept`** + never allocate memory on the heap

- `[[clang::nolock]]`

- **`noalloc`** + never block on a lock

- Can be inferred in some cases

- E.g. function body is visible

- Verified statically at compile time 😎

```
void noLockFunction() [[clang::nolock]];
void noAllocFunction() [[clang::noalloc]];

struct widget
{
    void noLockMethod() [[clang::nolock]];
};

void myFunction() [[clang::nolock]]
{
    noLockFunction();
    noAllocFunction(); // Error!

    widget w;
    w.noLockMethod();
}
```

noalloc, nolock

https://discourse.llvm.org/t/rfc-nolock-and-noalloc-attributes

- Similar to `noexcept`
  - `[[clang::noalloc]]`
    - **`noexcept`** + never allocate memory on the heap
  - `[[clang::nolock]]`
    - **`noalloc`** + never block on a lock
- Can be inferred in some cases
  - E.g. function body is visible
- Verified statically at compile time 😎

```
void noLockFunction() [[clang::nolock]];
void noAllocFunction() [[clang::noalloc]];

struct widget
{
    void noLockMethod() [[clang::nolock]];
};

void myFunction() [[clang::nolock]]
{
    noLockFunction();
    noAllocFunction(); // Error!

    widget w;
    w.noLockMethod();
}
```

	GUI (System Trace)	cli (dtrace)	code	interpose	RTSan
Easy to use?	✅/⚠️	⚠️	✅	⚠️	✅
Clear?	❌	✅	✅	✅	✅
🧵Filterable?	⚠️	⚠️	✅	✅	✅
CI?	❌	⚠️	✅	✅	✅
Portable?	❌	❌	✅	⚠️	⚠️
System calls?	✅	✅	❌	⚠️	⚠️
Malloc/free?	⚠️	✅	❌	✅	✅
Lock/unlock?	✅	✅	⚠️	✅	✅
3rd party code?	✅	✅	❌	✅	✅
Notes	Different tools for different tasks	Requires disabling SIP		No raw/inline context switches/syscalls	No raw/inline context switches/syscalls