

Contents

1	Module Aio : libaio-ocaml Linux async I/O interface for ocaml	1
2		1

1 Module Aio : libaio-ocaml Linux async I/O interface for ocaml

This module implements the libaio bindings that interface with the Linux system calls.

Version 0.0.0 - goswin-v-b@web.de

2

type `buffer`

The type for Buffer.

val `buffer` : `int` -> `buffer`

Allocate an uninitialized buffer.

val `get_string` : `buffer` -> `int` -> `string`

Get a string of length x from a buffer.

val `put_string` : `buffer` -> `string` -> `unit`

Put a string into a buffer.

val `rewind` : `buffer` -> `unit`

Rewind a buffer to it start.

type `result`

The type for a result of a completed I/O request

exception `Error` of `int`

An error has occured during a request.

exception `Incomplete` of `buffer * int`

A request was only partialy completed.

val `result` : `result` -> `buffer`

Extract the buffer from a result or throw the proper exception

type `context`

The type for a libaio Context.

val `context` : `int` -> `context`

Create a new context for n simultaneous requests.

```
val read :  
  context ->  
  Unix.file_descr -> buffer -> int64 -> (result -> unit) -> unit  
    fill buffer from file at given offset and call continuation  
  
val write :  
  context ->  
  Unix.file_descr -> buffer -> int64 -> (result -> unit) -> unit  
    write buffer to file at given offset and call continuation  
  
val run : context -> unit  
    run the context till there are no more pending requests
```