Operating Systems Lab (C+Unix)

Enrico Bini

University of Turin

Outline

Errors: the errno global variable

- The invocation of functions may fail. Examples:
  - `malloc` fails if memory is not available
  - `open` fails if the file to be read does not exist
- If the call to a function fails, the caller is informed by an invalid returned value. Examples:
  - `malloc` returns `NULL` if failing (invalid pointer)
  - `open` returns `-1` if failing (invalid file decriptor)
- If the invalid value is returned, the calling function knows that an error happened, but **doesn't know why**
- To inform about the cause of the error the global variable

```
int errno;
```

  is set by the failing function. It is available by including

```
#include <errno.h>
```

- When a function fails, it sets the global variable `errno` accordingly.
- The caller may get more details about the reasons of failure by inspecting the value of `errno`

Errors: values of `errno`

- The `man` pages of the failing function list all possible values of `errno` which may be set, in the section "`ERRORS`" usually at the bottom of the `man` page. Example: `man 2 open`
- These values are pre-processor macros set equal to non-zero integers.
- If `errno == 0`, then the previous call was successful
- the function (declared in `string.h`)

```
char * strerror(int errnum);
```

  returns a pointer to a string describing the error with code `errno`

- *`test-error-open.c`*