The Command Line, the Environment, and Time

Chris Brown

In This Module …

Accessing the command line

Processing command options

The environment

Time

Representations, conversions

Time zone and locale

Process times

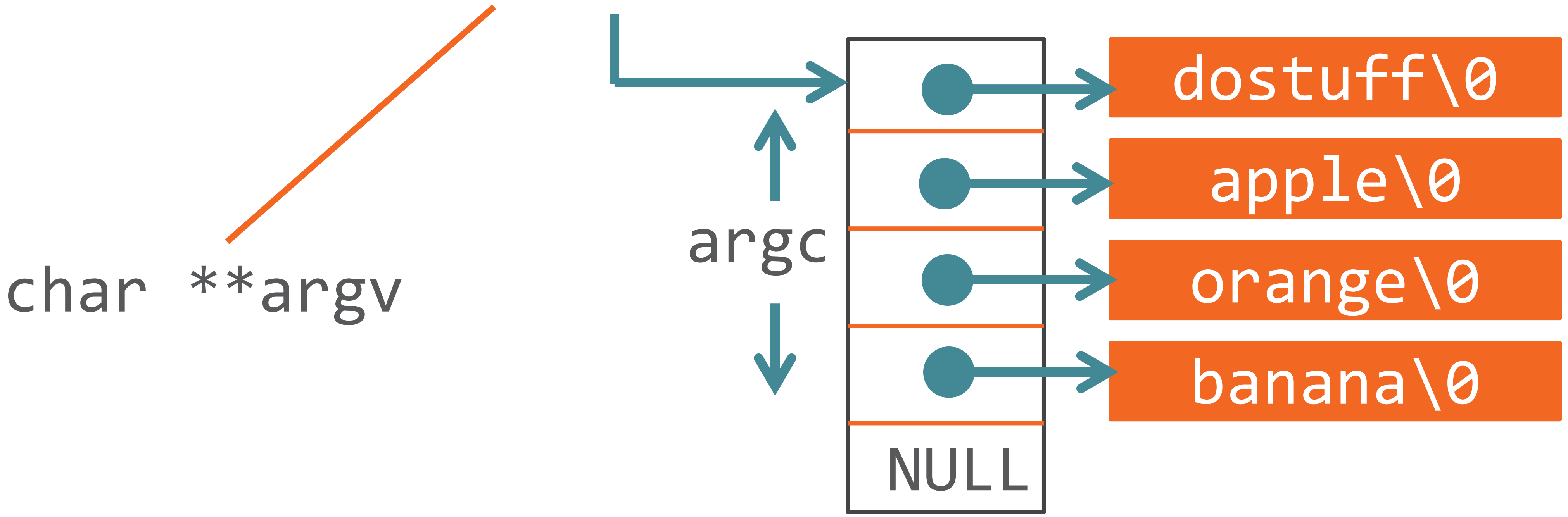
Demonstrations

Command Line Arguments

```
$ dostuff apple orange banana
```

Command line

```
int main(char *argv[], argc)
```



Traversing the Command Line

```
#include <stdio.h>

int main(int argc, char*argv[])
{
  int i;

  for (i=0; i< argc; i++)
    printf("%s\n", argv[i]);
}
```

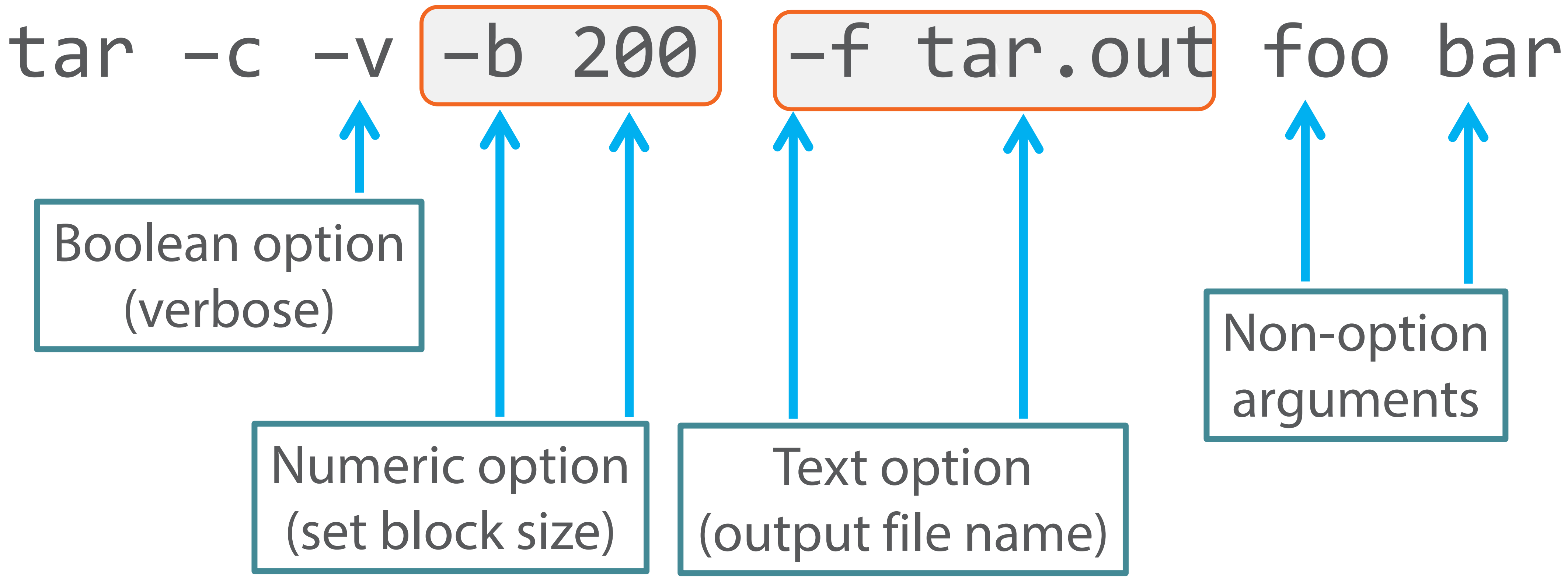
Usage Error Reporting

A program that expects exactly one argument

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  if (argc != 2) {
    fprintf(stderr, "usage: %s file\n", argv[0]);
    exit(1);
  }
  printf("processing %s\n", argv[1] );
}
```

Handling Command Options


```
tar -c -v -b 200 -f tar.out foo bar
```
Boolean option (verbose)
Numeric option (set block size)
Text option (output file name)
Non-option arguments

Combining Options

```
ls -l -a
```

Equivalent commands

```
ls -la
```

```
tar -cvf tar.out foo bar
```

Boolean and text options combined

It's all starting to get complicated!

Option Processing Using `getopt()`

```
getopt(argc, argv, optstring)
```

Returns the option letter (or EOF if there are no more)

`argc`: As passed to `main()`

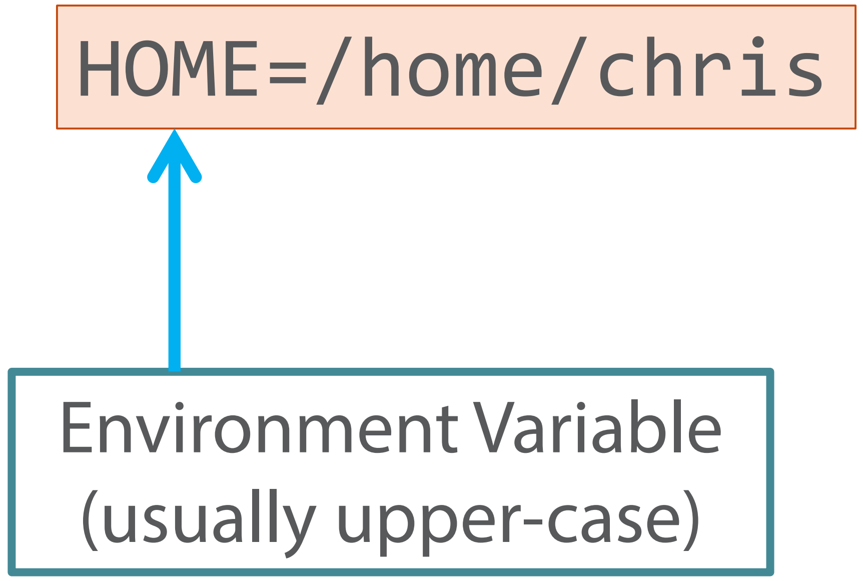
`argv`: As passed to `main()` On completion, `argv` is re-ordered to put all the options first

`optstring`: String specifying the allowed option letters. A following ":" says the option takes an argument

Example: `"abn:t:"`

The Environment

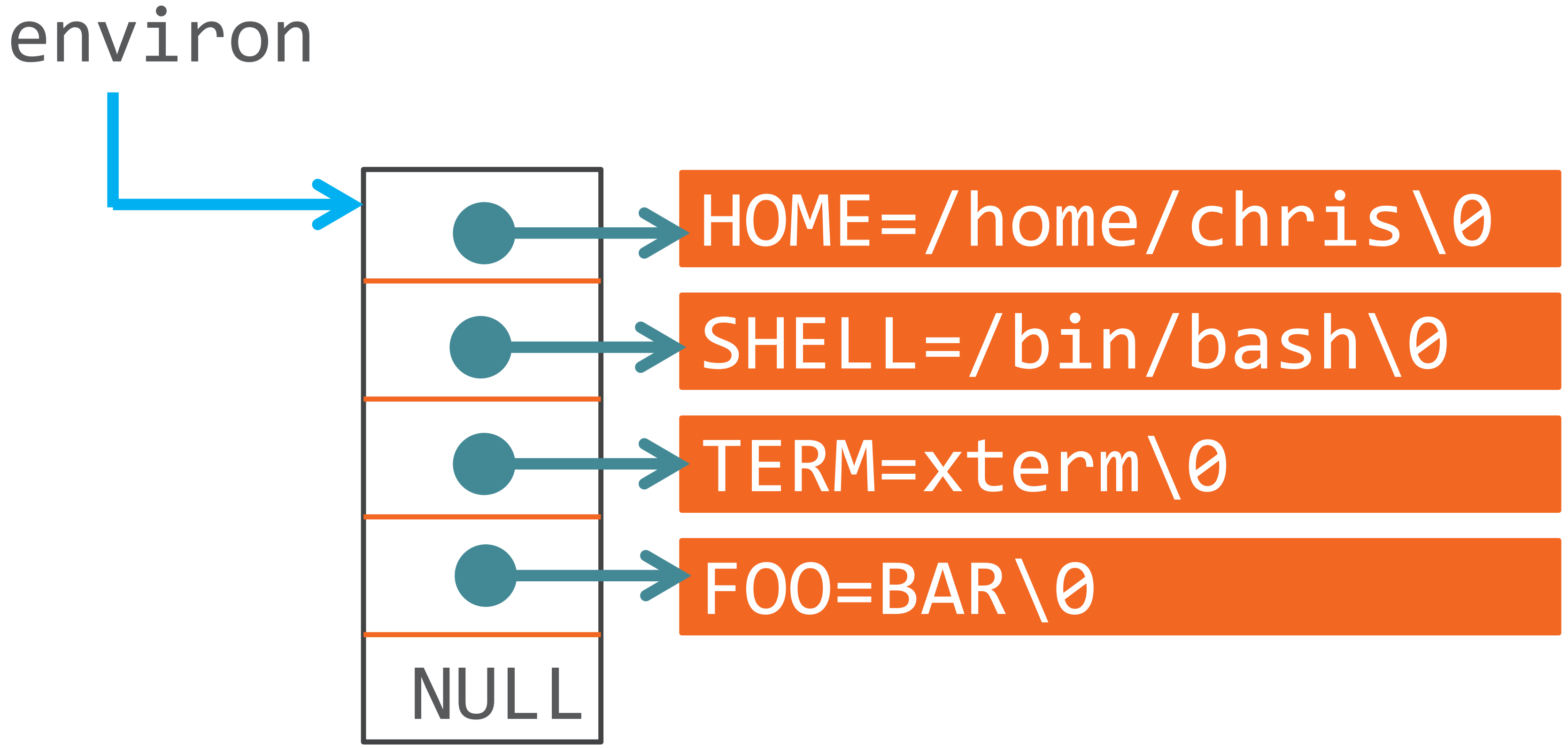
A list of strings carried by each process



On the command line:

```
$ export FOO=BAR
$ env | grep FOO
FOO=BAR
```

The Environment



Listing the Environment

```
#include <stdio.h>

extern char **environ;

void main(int argc, char *argv[])
{
  char **p;

  for (p=environ; *p != NULL; p++)
    printf("%s\n", *p);
}
```

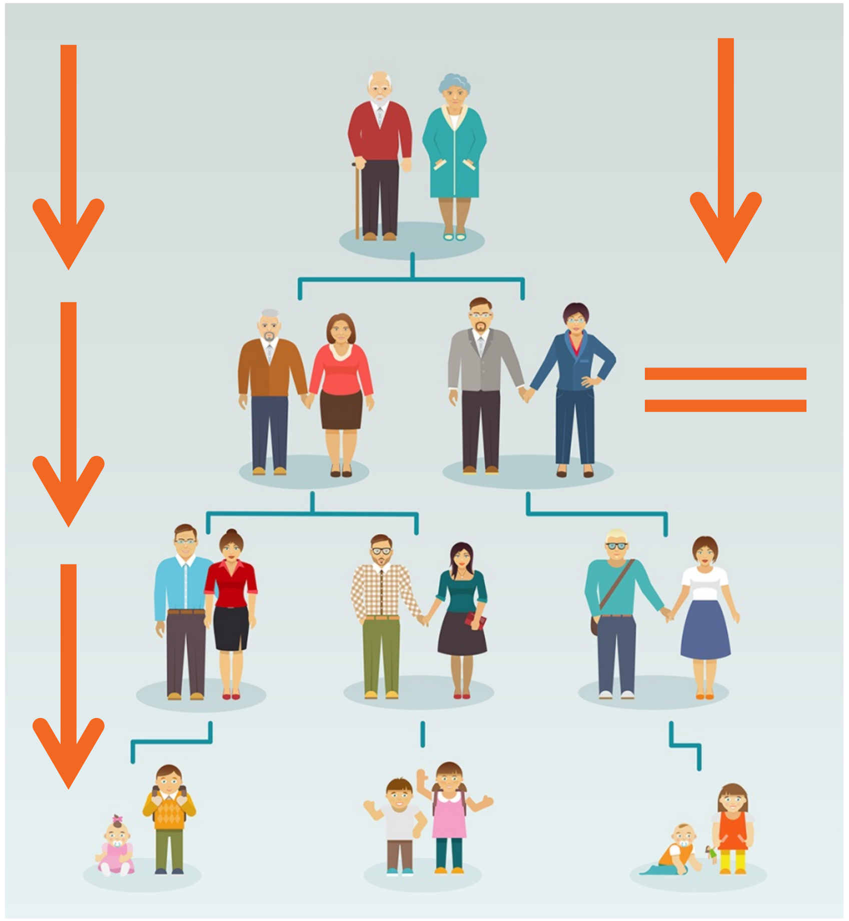
Querying the Environment

```
getenv(name)
```

Returns the associated string value or NULL if none

The environment variable's name

Inheriting the Environment



The environment is normally passed down from a process to its children

… and their children in turn

A process can choose *not* to pass on its environment

Time

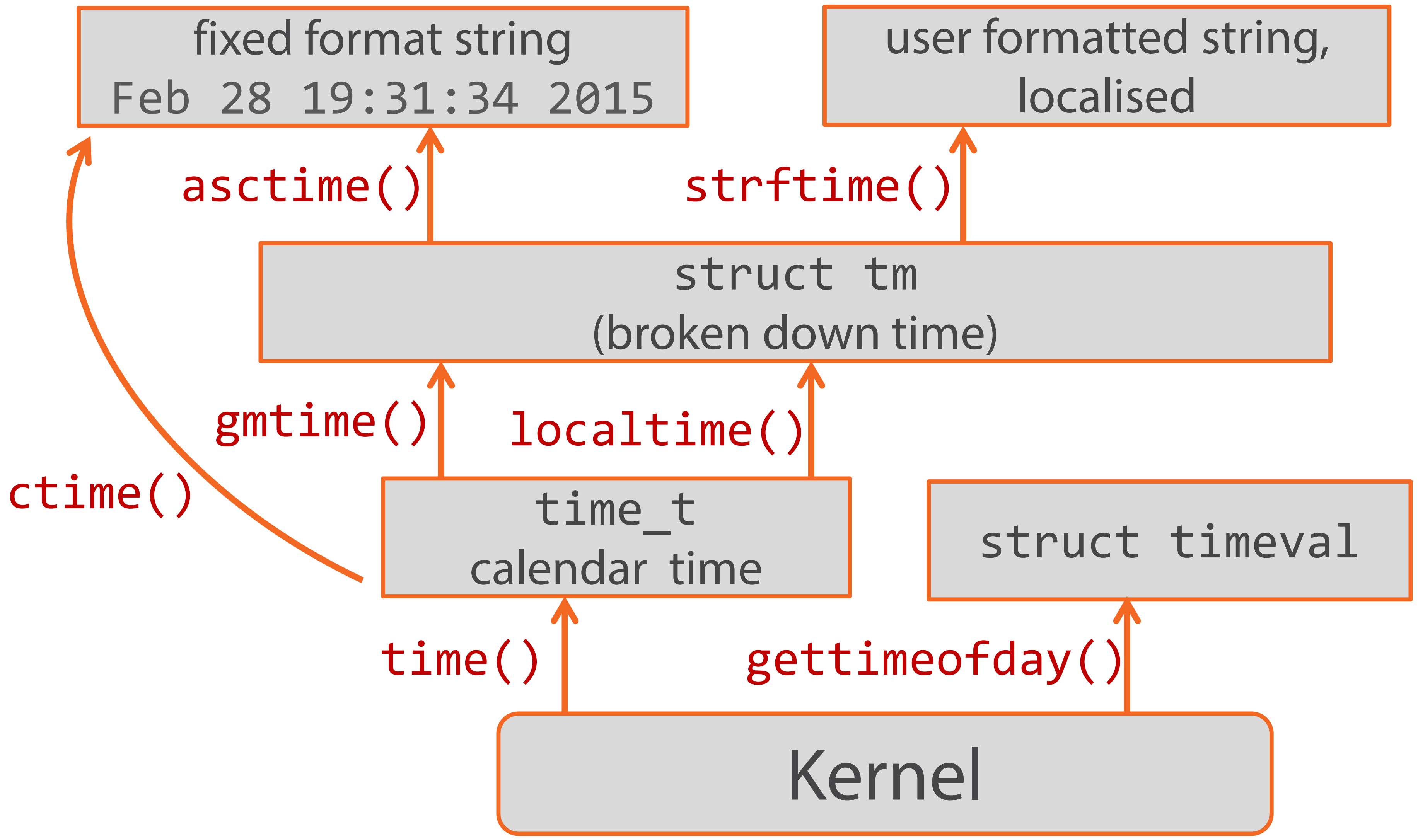
Representations of time

Conversions

Time zones and locales

Measuring process times

Time Conversions



What Time Is It?

`time(NULL)`

Returns the number of seconds since "the epoch" (Midnight, 1 Jan 1970, UTC)

From the command line:

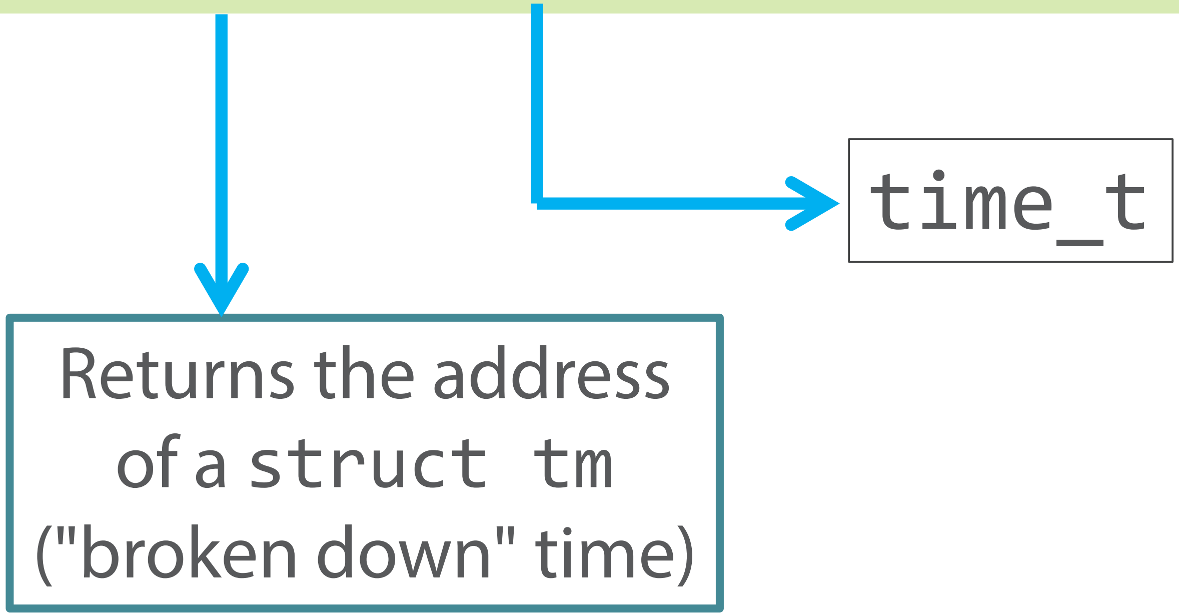
```
$ date +%s
1425812803
```

On a 32-bit system, a `time_t` will overflow in January 2038

Broken Down Time

`gmtime(t)` Time returned in UTC

`localtime(t)` Time returned in local time zone



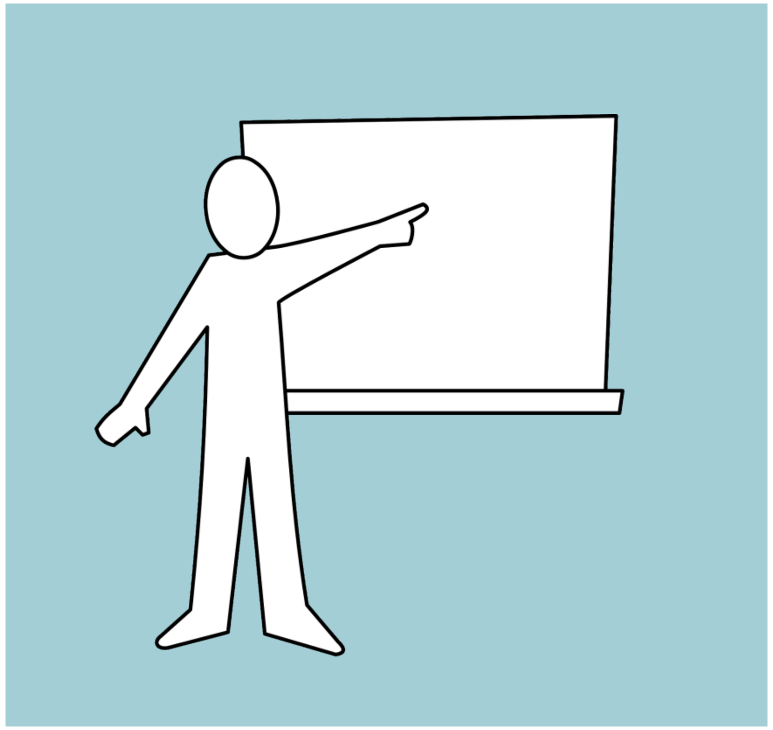
`time_t`

Returns the address of a `struct tm` ("broken down" time)

The tm Structure

```
struct tm {
     int tm_sec;          /* seconds */
     int tm_min;          /* minutes */
     int tm_hour;         /* hours */
     int tm_mday;         /* day of the month */
     int tm_mon;          /* month */
     int tm_year;         /* year */
     int tm_wday;         /* day of the week */
     int tm_yday;         /* day in the year */
     int tm_isdst;        /* daylight saving time */
};
```

Demonstration



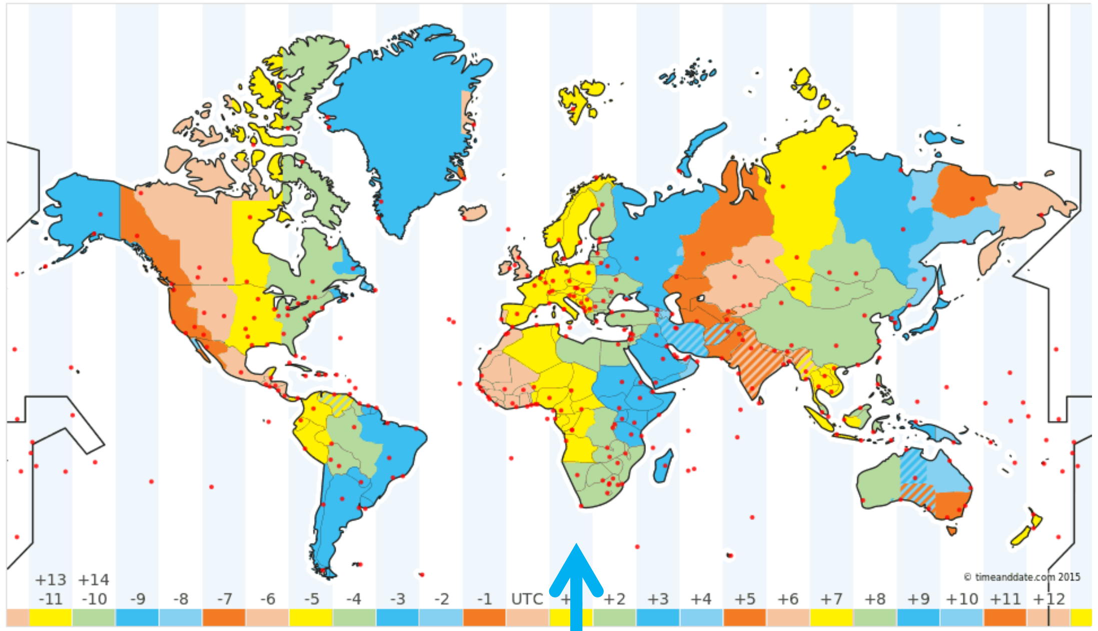
Recursive Directory Traversal
— invoked by –r option

Get "last modification" time of each file

Break down into hours, minutes etc.

Histogram based on hour

Time Zones



+13 +14
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 UTC +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12
© timeanddate.com 2015

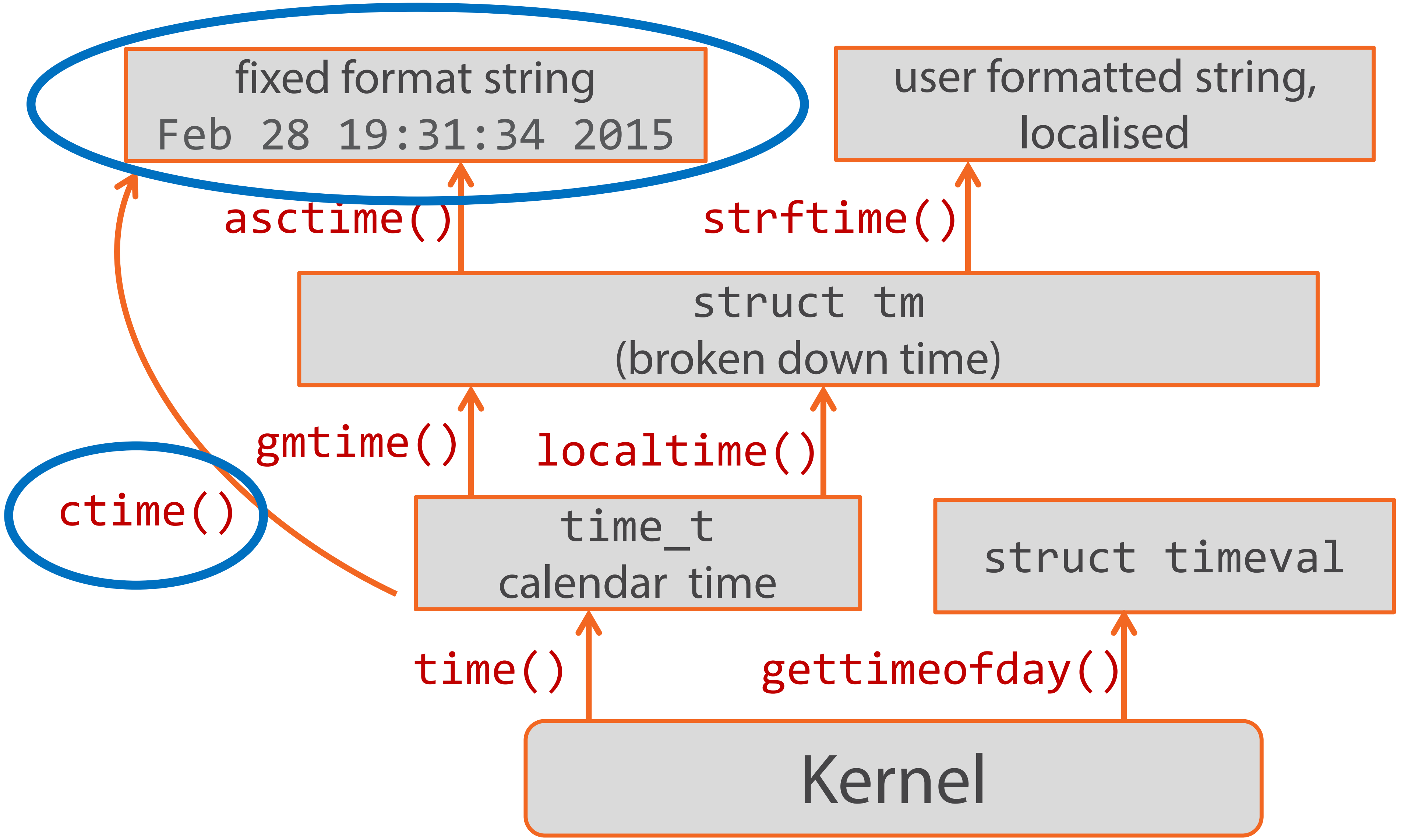
Many time conversion routines in `glibc` consult the environment variable TZ

… or failing that, use the file `/etc/localtime`

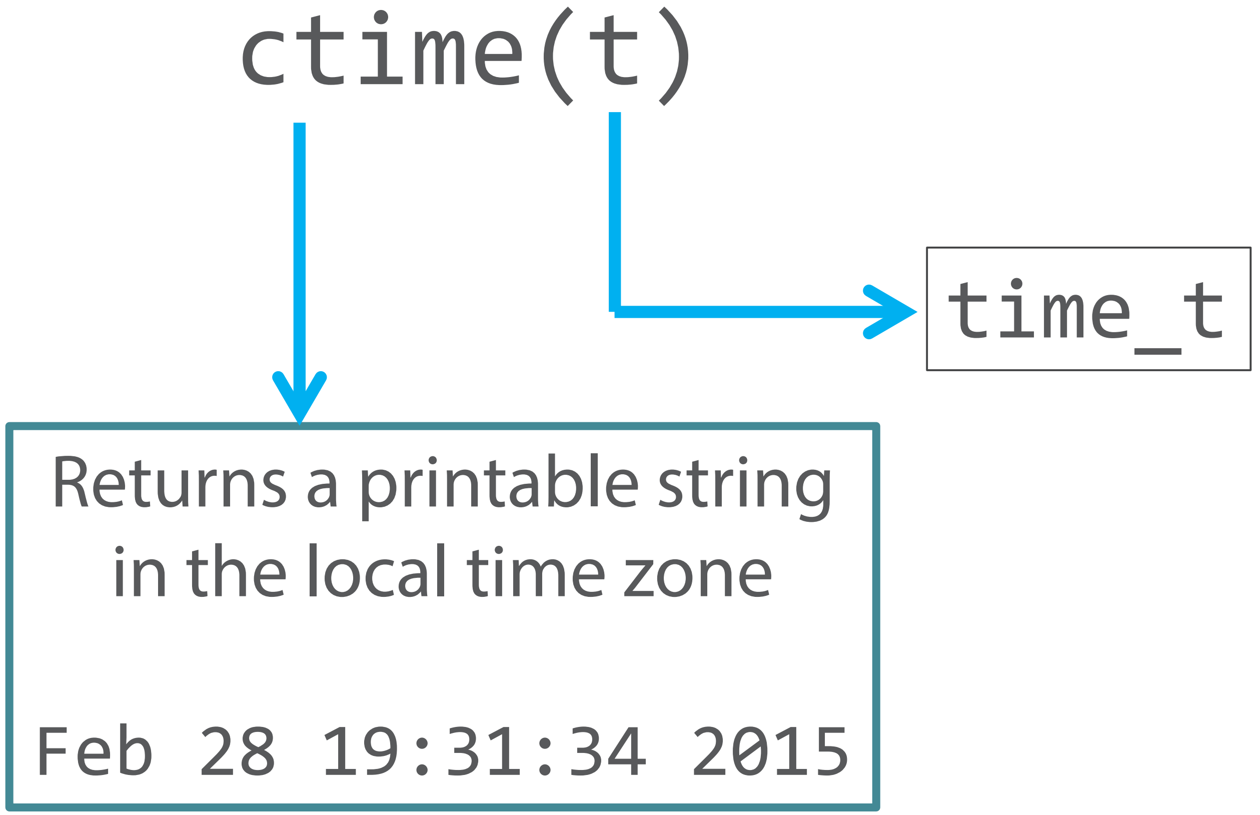
`TZ=":CET"`

Timezone info stored in `/usr/share/zoneinfo/CET`

Time Conversions



Converting to a Human-Readable Form



```
ctime(t)
```

```
time_t
```

Returns a printable string in the local time zone

```
Feb 28 19:31:34 2015
```

We Don't All Speak the Same Language!

Oslo
Gothenburg
Edinburgh
DENMARK
Copenhagen
Belfast
Dublin
IRELAND
UNITED KINGDOM
THE NETHERLANDS
Berlin
Amsterdam
Cardiff
London
GERMANY
Brussels
BELGIUM
Frankfurt
Prague
CZECH REPU
Paris
LUXEMBOURG
Rennes
Vienna
Munich
AUSTRIA
FRANCE
Bern
SWITZERLAND
SLOVENIA
Ljubliana
Lyon
Bordeaux
Milan
CROATI
Bilbao
Marseille
ITALY
Porto
Barcelona
Ajaccio
Rome
PORTUGAL
Madrid
SPAIN
Lisbon
Valencia
Palma
Naple
Cagliari
Cordoba

"Sunday 8 March"

"Sonntag 8 März"

"Dimanche 8 Mars"

Locales

- A *locale* defines conventions for displaying money amounts, times and dates and numbers
  - Defined by files under `/usr/share/locale`
- May need to install additional languages, e.g. on Ubuntu:
  - `sudo apt-get install language-pack-de`
- Specify the locale by setting the environment variable `LC_ALL`
  - `LC_ALL=de_DE.utf8; export LC_ALL`
- Make the program aware of the locale:
  - `setlocale(LC_ALL, "")`

Converting Time to a Locale-Specific String

```
strftime(buf, 1000, "%A %e %B", tm)
```

String returned in this buffer

Size of buffer

Format string. Codes include:

%y	2-digit year
%Y	4-digit year
%R	24-hour time
%b	Short month name
%B	Full month name

… see man page for full list

Pointer to `struct tm`

Measuring Process Time

`clock()`

Returns a clock_t giving elapsed process time in microseconds

This does *not* include process time of child processes

Measuring Process Time

```
times(buf)
```

$\to$ `struct tms`

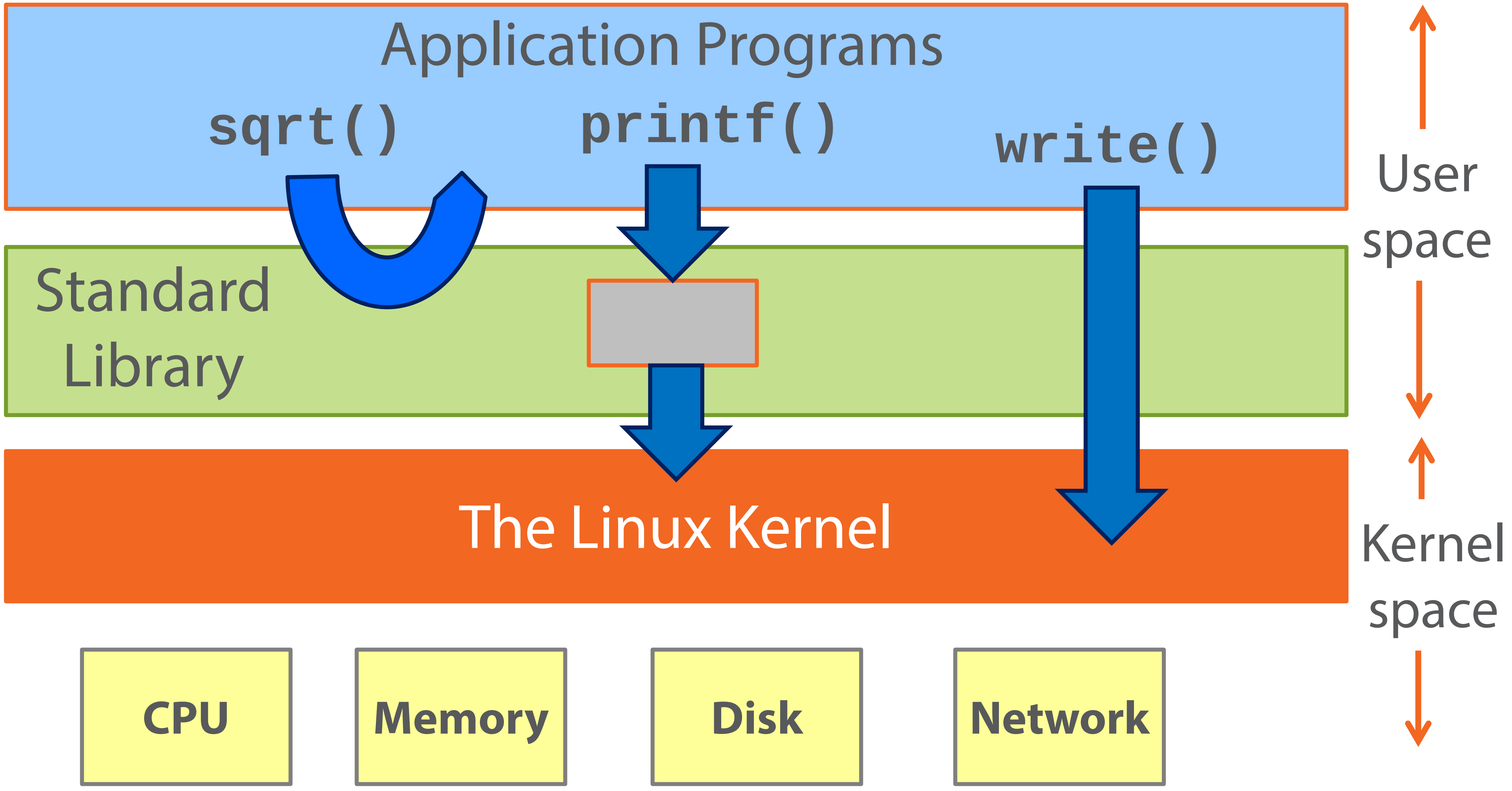
Clock frequency returned by sysconf(_SC_CLK_TCK) (probably 100 Hz)

Returns user time and system time for the process and for its terminated children

The tms Structure

```
struct tms {
  clock_t tms_utime;  /* user time */
  clock_t tms_stime;  /* system time */
  clock_t tms_cutime; /* user time of children */
  clock_t tms_cstime; /* system time of children */
};
```

Kernel Space and User Space



Module Summary

Command line arguments

- Option processing with `getopt()`

The environment

Time

- Representations (`time_t`, broken-down time)
- Conversions
- Printable representations, timezones, locales
- Process times (system time and user time)

Coming up in the Next Module

Processes

Pipes