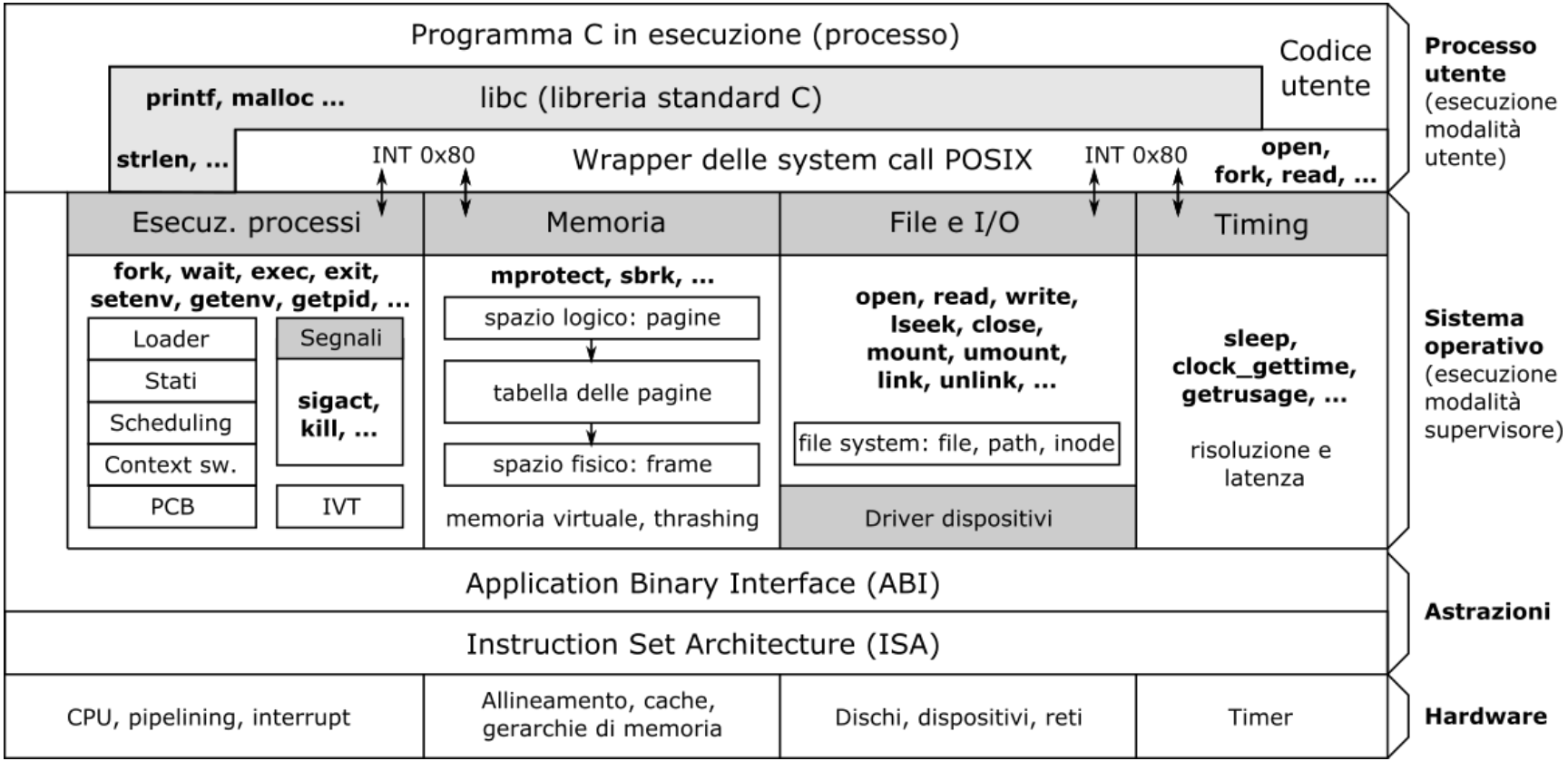
SAPIENZA
UNIVERSITÀ DI ROMA

Sistemi di calcolo

Capitolo 4
Librerie standard per il programmatore C

Corso di Laurea in Ingegneria Informatica e Automatica

Hardware-sofware stack di un sistema Linux



Accesso alla documentazione: man

DEMO: man printf, man write, man strlen

Gestione degli errori

Gestore terminazione di un programma: atexit

```
#include <stdlib.h>
int atexit(void (*function)(void));
```

Parametri:

- `function`: puntatore a una funzione da eseguire in caso di terminazione con la funzione `exit` oppure con `return` dal `main`

Risultato:

- 0 in caso di successo, -1 in caso di errore

Gestore terminazione di un programma: atexit

Esempio:

```
#include <stdio.h>   // printf
#include <stdlib.h>  // atexit

void handler() {
    printf("uscita dal programma\n");
}

int main() {
    atexit(handler);
    return 0; // stampa: "uscita dal programma"
}
```

Errori non recuperabili

```
#include <assert.h>
assert(test);
```

Parametri:

- `test`: valore di verità. Se diverso da zero, provoca la terminazione del programma con un codice diverso da zero e invia sul canale stderr un messaggio di errore della forma "Assertion failed: (<test>), function main, file <nomefile.c>, line <numero-linea>"

Risultato:

- nessuno

Errori recuperabili

Pattern gestione errori recuperabili

```
int mia_funzione() {
    int res, *buffer = NULL;
    ...
    buffer = malloc(...);
    if (buffer == NULL) goto cleanup; // errore nella chiamata
    ...
    res = chiamata(...);
    if (res != 0) goto cleanup;       // errore nella chiamata
    ...
    free(buffer);
    return 0;

cleanup:
    ...
    if (buffer != NULL) free(buffer); // deallocazione buffer
    return -1;
}
```

Libreria standard C: gestione delle stringhe

```
#include <string.h>

size_t strlen(const char *s);

char *strcpy(char *dest, const char *src);

char *strcat(char *dest, const char *src);

int strcmp(const char *s1, const char *s2);

char *strtok(char *str, const char *delim);


#include <stdlib.h>
int atoi(const char *nptr);
```

Libreria standard C: gestione delle stringhe

DEMO 4.1-strings

Libreria standard C: gestione delle stringhe

```
#include <stdio.h>

int sprintf(char *str, const char *format, ...);

int sscanf(char *str, const char *format, ...);
```

Libreria standard C: gestione delle stringhe

DEMO 4.1-strings

Libreria standard C: manipolazione memoria

```
#include <string.h>

void *memcpy(void *dest, const void *src, size_t n);

int memcmp(const void *s1, const void *s2, size_t n);

void *memset(const void *dest, int c, size_t n);
```

Libreria standard C: manipolazione memoria

DEMO 4.2-mem

Libreria standard C: stdin, stoud, stderr

Libreria standard C: gestione dei file di testo

```
#include <stdio.h>

FILE *fopen(const char *pathname, const char *mode);

int fclose(FILE *stream);


int fprintf(FILE *stream, const char *format, ...);

int fscanf(FILE *stream, const char *format, ...);

char* fgets(char *str, int size, FILE *stream);
```

Libreria standard C: gestione dei file binari

```
#include <stdio.h>

size_t fwrite(const void *ptr, size_t size,
              size_t nitems, FILE *stream);

size_t fread(void *ptr, size_t size,
             size_t nitems, FILE *stream);



int fseek(FILE *stream, long offset, int whence);

long ftell(FILE *stream)
```

Libreria standard C: file

DEMO 4.3-file

Libreria standard C: ordinamento e ricerca

```
#include <stdlib.h>

void qsort(void *base, size_t nel, size_t width,
     int (*compar)(const void *, const void *));

void *bsearch(const void *key, const void *base,
     size_t nel, size_t width,
     int (*compar) (const void *, const void *));
```

Comprensione espressioni di tipo in C

Comprensione espressioni di tipo in C

Libreria standard C: ordinamento e ricerca

DEMO 4.4-sort-search

Ordinamento di interi

```
#include "e1.h"
#include <stdlib.h>
#include <string.h>

int compar(const void* a, const void* b) {
    int x = *(int*)a;
    int y = *(int*)b;
    return x-y;
}

void sort_ints(int ints[], size_t size){
    qsort(ints, size, sizeof(int), compar);
}
```

Ordinamento di stringhe

```
#include "e1.h"
#include <stdlib.h>
#include <string.h>

int compar(const void* a, const void* b) {
    const char* x = *(char**)a;
    const char* y = *(char**)b;
    return strcmp(x, y);
}

void sort_strings(char *strings[], size_t size){
    qsort(strings, size, sizeof(char*), compar);
}
```

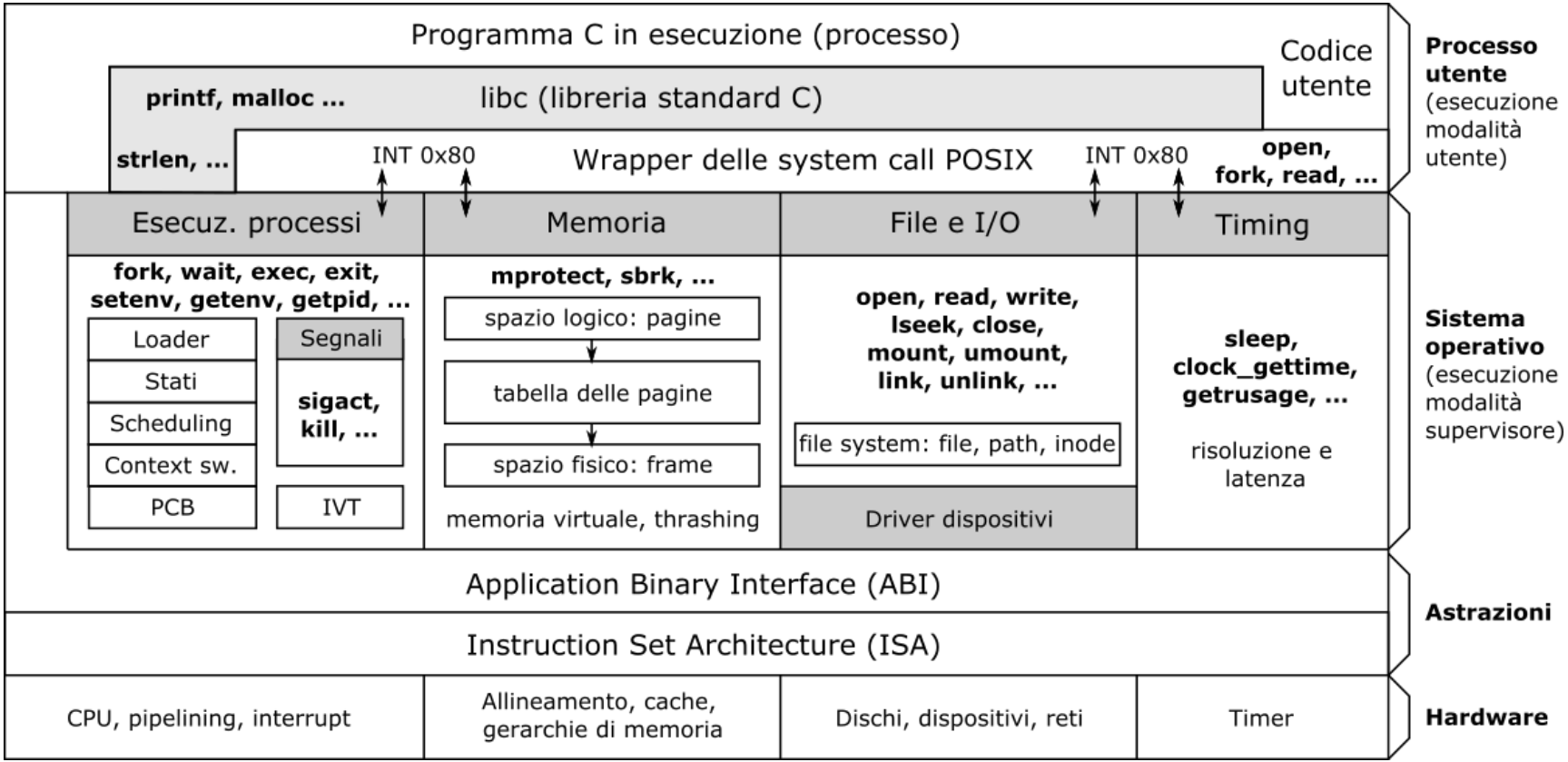
Libreria standard C: funzioni e costanti matematiche

```
#include <math.h>

double sqrt(double);
double log(double);
double sin(double);
double cos(double);
double tan(double);
...
```

https://pubs.opengroup.org/onlinepubs/9699919799/basedefs/math.h.html

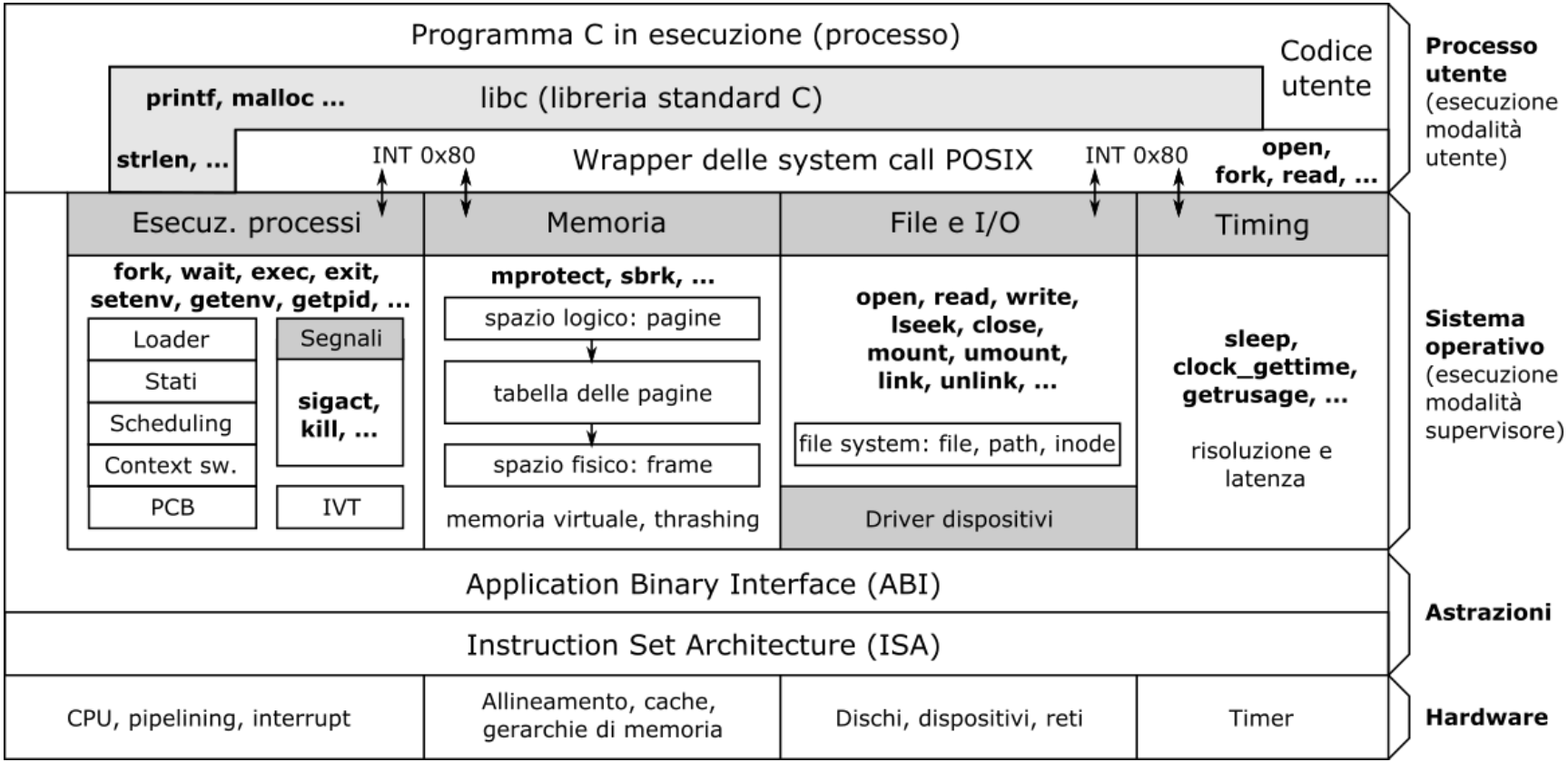
Tracciamento system call: strace



Tracciamento system call

DEMO 4.5-strace

Tracciamento chiamate a librerie: ltrace



Tracciamento chiamate a librerie

DEMO 4.5-ltrace