Laboratorio.

Per poter realizzare un tipo di programmazione “di sistema”, cioè a più basso livello, bisogna passare per metodi non ortodossi.

Donald Knuth ha scritto “The art of computer programming”. Egli ha anche sviluppato un’applicazione particolare, TEX, un word processor abbastanza evoluto. Egli lo ha sviluppato per rendere gradevoli alla vista i documenti formattati. Lo scopo per cui la sviluppò era per stampare i sorgenti dei programmi in tutto il loro lato artistico.

Un algoritmo può essere provato corretto, ma la sua implementazione può non esserlo: l’unico modo per accorgersi della presenza di questi errori è di mandare in esecuzione il programma.

Gli esercizi di programmazione delle sessioni di laboratorio sono da sviluppare in ambiente POSIX con il linguaggio di programmazione C. POSIX fa riferimento a UNIX, un sistema operativo molto significativo per l’epoca sviluppato da un piccolo gruppo di persone, in particolare a Ken Thompson. Egli, insieme a Dennis Ritchie, era pagato da un’azienda telefonica americana per fare quello che piaceva più a loro: loro hanno deciso di sviluppare un sistema operativo. La differenza è che questo sistema operativo, anziché essere programmato in assembly (come gli altri sistemi di quel tempo) era programmato in C: questo portava a diversi vantaggi, quali la compattezza e mantenibilità del codice (le centinaia e migliaia di istruzioni degli altri sistemi erano significativamente ridotte in poche decine di migliaia di righe di codice in linguaggio di alto livello). Successivamente il sistema UNIX ha avuto una sua evoluzione: diversi gruppi di persone hanno sviluppato la propria versione partendo dalla versione di Thompson e Ritchie. Il vantaggio del linguaggio C è che sebbene sia un linguaggio di alto livello, permette di scendere di livello fino ad interagire direttamente con le rappresentazioni binarie delle informazioni.

Dopo una varietà di versioni diverse si è stabilito un obiettivo di standardizzazione per facilitare la portabilità dei sistemi operativi: questo standard è POSIX. Seguendo lo standard POSIX si possono sviluppare applicazioni capaci di girare su tutte le versioni di UNIX. POSIX ad oggi lo troviamo su Linux e anche su Windows (Microsoft ha dovuto cedere), MacOS e in generale tutte le piattaforme disponibili per la programmazione.

I nostri esercizi di programmazione saranno basati sulle system call POSIX e sull’uso del linguaggio C. Pei il linguaggio C esistono diversi compilatori, il più affermato è il compilatore della GNU: gcc. Una componente molto importante del nostro ambiente di sviluppo è la shell: essa è l’interfaccia di programmazione testuale; i suoi vantaggi rispetto a un editor normale è che può essere utilizzata sia in modalità interattiva, che in modalità script (una serie di comandi di shell possono essere scritti all’interno di un file che possono essere interpretati da una shell bash).

Vedremo come produrre archivi attraverso il comando tar, e il sistema di compressione gzip, che crea file tgz (tar gzip).   
Si possono creare eseguibili attraverso il comando “make”: una delle caratteristiche del C è quella di poter suddividere il sorgente in più file da compilare separatamente e poi di poter unire i codici oggetto attraverso il linking. Il “make” chiama nella giusta sequenza il programma di compilazione sui file sorgenti.

La nostra applicazione presenta un makefile: il suo contenuto è la descrizione di quel che deve fare la funzione make, secondo cui essa è in grado di produrre le applicazioni.

Altre componenti fondamentali sono il debugger (gdb), anche se non è strettamente indispensabile (si può evitare di usare il debugger e sostituirlo con delle scritte che indichino sul terminale il valore delle nostre variabili “critiche” durante l’esecuzione.

C’è poi la libreria delle funzioni predefinite: glibc (esse possono essere usate per una grande varietà di casi).

Come text editor si può utilizzare vim.

Infine, ma di grande importanza, è il manuale (man): esso è il riferimento che descrive tutte le cose fondamentali da sapere per una programmazione in ambiente POSIX. Per usarlo basta scrivere in una shell man seguito dalla parola di cui si vuole scoprire di più (per esmpio si può chiamare man man).