

# Operating Systems Lab (C+Unix)

**Enrico Bini**

University of Turin

# Outline

## 1 Unix: introduction and basic usage

- Shell
- File system
- Accounts

- Enrico Bini

<http://www.di.unito.it/~bini/>

- Areas di ricerca:

- ▶ embedded systems
- ▶ operating systems (Linux scheduler and tracing)
- ▶ control systems (UAVs)
- ▶ resource optimization over IoT/edge/cloud computing

# Divisione degli studenti in Turni

- L'insegnamento di Sistemi Operativi è suddiviso in due parti:
  - ▶ teoria: ore "SO" in orario
  - ▶ laboratorio: ore "SO lab C" + ore "SO lab Unix" in orario Durante tutte le ore di laboratorio si insegnerà
    - ★ Programmazione in C (fino a circa fine Ottobre)
    - ★ Programmazione concorrente con Unix system calls (dopo il C)
- Voi studenti siete divisi in **due** corsi sulla base dell'iniziale del cognome. I docenti di teoria sono:
  - ▶ **Corso A**: iniziale da "A" a "K": Gunetti
  - ▶ **Corso B**: iniziale da "L" a "Z": Baroglio
- Gli studenti sono anche divisi in **quattro** turni di laboratorio, denotati da  $Tx$  con  $x \in \{1, 2, 3, 4\}$ , sulla base del numero di matricola. I docenti assegnati a ciascun turno sono:
  - ▶ **T1** = corso A e matricola dispari: De Pierro (C) e Radicioni (Unix)
  - ▶ **T2** = corso A e matricola pari: **Bini** (C+Unix)
  - ▶ **T3** = corso B e matricola dispari: **Bini** (C+Unix)
  - ▶ **T4** = corso B e matricola pari: De Pierro (C) and Schifanella (Unix)

## FAQ: “Posso seguire lezioni degli altri turni?”

- La divisione degli studenti fra turni esiste per bilanciare il lavoro dei docenti e la capacità delle aule. Quindi, seguire le lezioni degli altri turni **non è consentito**.
- Rischi nel seguire le lezioni di altri docenti:
  - ① possibilità di perdere qualche argomento a causa del diverso calendario dei vari turni
  - ② ricevere una valutazione da un docente diverso da quello con cui si è seguito: la discussione del progetto e la correzione dello scritto vengono comunque fatte dal vostro docente di turno “ufficiale”

# Esame del corso

## 1 Progetto

- ▶ **necessario** per essere ammessi allo scritto
- ▶ oggetto del progetto verrà illustrato dal docente “Unix”:  
Radicioni (T1), **Bini** (T2+T3), Schifanella (T4)
- ▶ progetto 2021 (scorso anno) si può discutere fino al 30 Novembre 2022
- ▶ dal 1 Dicembre 2022 si discute il progetto 2022 (anno corrente)
- ▶ ogni studente riceve una votazione **individuale** in 30-esimi, basata su
  - ★ qualità del progetto e contributo personale ad esso
  - ★ conoscenza degli argomenti trattati nelle ore di Laboratorio
- ▶ la votazione è mantenuta per i **cinque appelli successivi** alla discussione

## 2 Scritto

- ▶ **obbligatorio**
- ▶ ammissione allo scritto è possibile soltanto se il progetto è stato preventivamente discusso con successo (votazione progetto  $\geq 18$ )
- ▶ contenuto: teoria (20 punti) e programmazione C (10 punti)
- ▶ votazione in 30-esimi

## 3 Voto finale

- ▶  $\frac{1}{4}(\text{progetto}) + \frac{3}{4}(\text{scritto})$
- ▶ possibilità di sostenere orale: sentire i docenti di teoria

# Progetto: oggetto

- Il progetto richiederà l'utilizzo di processi per la programmazione concorrente
- Sviluppato in C
- Numerosi studenti che hanno superato l'esame hanno dichiarato:
  - ▶ “il progetto di Unix è mooolto più difficile dei progettini del primo anno”
  - ▶ “se avessi saputo dall'inizio che il progetto era così impegnativo, sarei stato più attento alle lezioni”
- Se si intende sostenere l'esame a Gennaio, si suggerisce di non perdere il passo delle lezioni
- Una soluzione progettuale degli anni passati è pubblicat sulla piattaforma i-learn

# Modalità di erogazione del corso

- [link a **pagina moodle**] Laboratorio di Sistemi Operativi: corso A (turno T2), corso B (turno T3)
- Pagina divisa per argomenti:
  - 1 "Installazione di Linux"
  - 2 "Unix: introduction and basic usage"
  - 3 ...
- Orario del corso: disponibile calendario Google gestito dal docente:
  - ▶ Calendario lezioni turno T2
  - ▶ Calendario lezioni turno T3
- Lezioni frontali come da orario
- Durante le lezioni frontali, fino al 26/9 sarà possibile anche seguire collegamento WebEx con studenti da remoto (per i link, si veda in calendario)
- Video-lezioni
- Forum del corso moodle. Utilizzare forum
  - ▶ sia per aspetti generali del corso
  - ▶ che per domande specifiche di qualche argomento (un forum per argomento)
  - ▶ sentitivi liberi di aiutare i vostri colleghi, rispondendo. Se serve il



# Materiale del corso

- Suddiviso per argomenti su moodle
- Slides (in Inglese)
  - ▶ A fine anno le slides verranno trasformate in documento in formato A4 per semplificarne la stampa. Vostri colleghi degli anni passati hanno dichiarato che era molto utile. Pubblicata versione scorso anno (ci potranno essere delle variazioni dell'ordine del 5%)
- Video-lezioni
- man pages (pagine di manuale in-linea)
- Programmazione C
  - ▶ Linguaggio C, fasi della compilazione, makefile  
Facchinetti, Larizza, Rubini, "Programmare in C"
  - ▶ Approfondimento del C su puntatori (argomento considerato ostico)  
Cabodi e altri, "Puntatori e strutture dati dinamiche"
  - ▶ [cppreference.com](http://en.cppreference.com) (anche in Italiano!)  
<http://en.cppreference.com/w/c>
  - ▶ The GNU C Library  
[https://www.gnu.org/software/libc/manual/html\\_node/](https://www.gnu.org/software/libc/manual/html_node/)
- Programmazione Unix
  - ▶ Dispensa in Italiano (di altri colleghi)

## Materiale del corso: uso di Google, stackexchange, ...

- Nella programmazione (e, ahime, in tanti altri ambiti), è consuetudine consultare “la rete”: Google, <https://stackoverflow.com>, ...
- Tali strumenti sono utili per una **prima risposta approssimativa**
- Il risultato delle ricerca deve poi essere approfondito e compreso individualmente utilizzando altre fonti
  - ▶ man pages
  - ▶ riferimenti forniti in precedenza
- Esempio di discussione del progetto: alla domanda “Perché hai invocato open con questa opzione?”, non è positivo rispondere:
  - ▶ “Perché era scritto così su stackexchange”
  - ▶ “Se la toglievo non funzionava”
  - ▶ “Perché me lo ha suggerito l'editor di testo”
  - ▶ “Perché me lo hanno detto i colleghi”vengono valutate negativamente.
- Questo non vuol dire dissimulare l'utilizzo della “rete” (“stackexchange? Mai sentito dire!”), ma essere in grado di capire individualmente quanto trovato
- Obiettivo del corso: imparare **NON** far funzionare

# Contenuti del corso

- Programmazione in C

- ▶ Pur essendo a 50 anni dalla sua nascita (1972), il C è il linguaggio standard per
  - ★ sistemi embedded (automotive, avionics, industrial automation, ...)
  - ★ sistemi operativi (dagli smartphone ai data center)
- ▶ Un buon programmatore in C trova lavoro facilmente presso: IBM, Google, SuSE, RedHat, ARM, Xilinx, Intel, etc. (per esperienza)
  - ★ Linux kernel developer
  - ★ device driver developer
  - ★ IoT/embedded systems programmer
  - ★ FPGA/custom hardware
  - ★ in generale, per la programmazione a basso livello (no pagine web)

# Outline

- 1 Unix: introduction and basic usage
  - Shell
  - File system
  - Accounts

# Istallazione di Linux

- Iniziamo con un bel quizzone!! (feedback anonimo utile al docente per comprendere il livello attuale della classe)
- Continuiamo poi con il materiale dell'argomento "Istallazione di Linux"