

```
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
```

```
#define MAXPAROLA 30
#define MAXRIGA 80
```

```
int main(int argc, char *argv[])
{
    int freq[MAXPAROLA]; /* vettore di contatori
delle frequenze delle lunghezze delle parole */
    char riga[MAXRIGA];
    int i, inizio, lunghezza;
    FILE *f;
```

```
for(i=0; i<MAXPAROLA; i++)
    freq[i]=0;
```

```
if(argc != 2)
```

```
{
    printf(stderr, "ERRORE, serve un parametro con il nome del file\n");
    exit(1);
}
```

```
f = fopen(argv[1], "r");
if(f==NULL)
```

```
{
    printf(stderr, "ERRORE, impossibile aprire il file %s\n", argv[1]);
    exit(1);
}
```

```
while( fgets( riga, MAXRIGA, f ) != NULL )
```



Sistemi Operativi

Introduzione al corso di Sistemi Operativi

Stefano Quer

Dipartimento di Automatica e Informatica

Politecnico di Torino

License Information

This work is licensed under the license



Attribution-NonCommercial-NoDerivatives 4.0 International

This license requires that reusers give credit to the creator. It allows reusers to copy and distribute the material in any medium or format in unadapted form and for noncommercial purposes only.

① **BY:** Credit must be given to you, the creator.

② **NC:** Only noncommercial use of your work is permitted.

Noncommercial means not primarily intended for or directed towards commercial advantage or monetary compensation.

③ **ND:** No derivatives or adaptations of your work are permitted.

To view a copy of the license, visit:

<https://creativecommons.org/licenses/by-nc-nd/4.0/?ref=chooser-v1>

Informazioni generali

- ❖ Sistemi Operativi
 - SSD: Ingegneria Informatica e dei Sistemi (ING-INF/05)
 - 05CJC_{OA}
- ❖ Laurea in Ingegneria Informatica (INF1T3)
- ❖ Anno 3, semestre 1
- ❖ 6 crediti, 60 ore
- ❖ Primo corso
 - Studenti da AAA a GIN
 - Studenti in debito esami

Numero studenti totali iscritti

A.A. 2018-2019 – 294
A.A. 2019-2020 – 267
A.A. 2020-2021 – 282
A.A. 2021-2022 – 272
A.A. 2022-2023 – 263
A.A. 2023-2024 – 191
A.A. 2024-2025 – 180

Istruttori



**Politecnico
di Torino**

Department of Control and
Computer Engineering



Cardone Lorenzo
lorenzo.cardone@polito.it



Quer Stefano
011-090-7076
stefano.quer@polito.it



Foscale Tommaso
tommaso.foscale@polito.it

Filipponi Gabriele
gabriele.filipponi@polito.it



Scanzio Stefano
011-090-5438
stefano.scanzio@polito.it



Risultati attesi

- ❖ Il modulo ha lo scopo di
 - Introdurre le caratteristiche, l'architettura e le funzionalità di base di un sistema operativo
 - Fornire conoscenze specifiche dei sistemi operativi UNIX-like (Linux)
 - Amministrazione di sistema, shell, comandi, filtri e linguaggi di script
 - Approfondire i concetti e le tecniche di programmazione concorrente
 - Gestione di processi e di thread
 - Problemi classici di sincronizzazione

Contenuto

❖ Argomenti principali

- Introduzione ai sistemi operativi
- File system (file, direttori, dischi)
- Processi (basi, controllo, segnali, pipe, etc.)
- Thread (basi, libreria pthread, etc.)
- Sincronizzazione (s/w, h/w, semafori, etc.)
- Scheduling dei processi
- Condizioni di stallo (deadlock)
- Ambiente UNIX/Linux
 - Comandi e amministrazione del sistema
 - Shell (interprete comandi UNIX)
 - Linguaggi di scripting (bash)

Argomento
distribuito durante
il corso

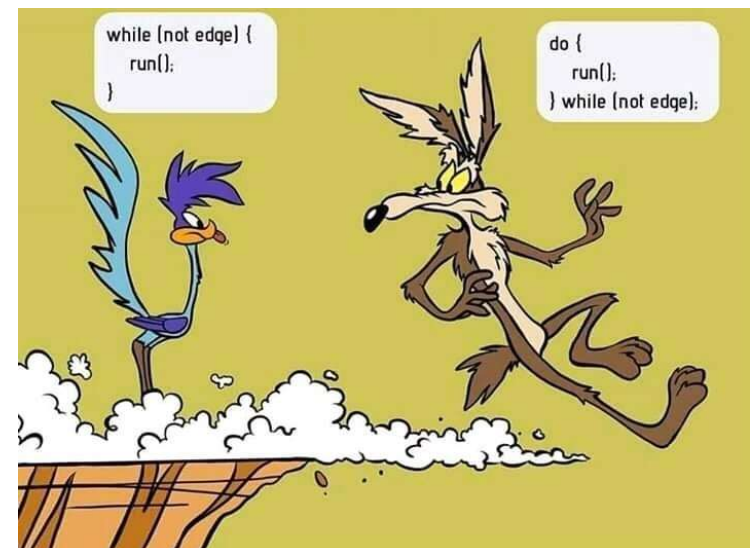
Prerequisiti

❖ Problem-solving tramite linguaggio C

- Conoscenza della sintassi (base e avanzata) del linguaggio di programmazione C

e.g., stringhe, puntatori, allocazione statica e dinamica di vettori e matrici, ricorsione, etc.

- Capacità di risolvere problemi di media complessità tramite la scrittura di programmi



Prerequisiti

- ❖ Conoscenza relative all'architettura di un sistema di elaborazione
 - Struttura di un processore, organizzazione della memoria, tipi di dato, etc.
 - Meccanismi di interruzione e trap

Impatto sulla carriera

❖ Impatto del modulo nel curriculum?



Preparing for Google Technical Internship Interviews

This guide is intended to help you prepare for Software Engineering internship and Engineering Practicum internship interviews at Google. If you have any additional questions, please don't hesitate to get in touch with your recruiter.



[Recruitment Process:
Engineering
Practicum Internships](#)



[Recruitment
Process: Software
Engineering
Internships](#)



[Interview Tips](#)



[Technical Preparation](#)



[Extra Prep
Resources](#)

Impatto sulla carriera

Google | Preparing for your Interview

Technical Preparation

Graphs: To consider a problem as a graph is often a very good abstraction to apply, since well known graph algorithms for distance, search, connectivity, cycle-detection etc. will then yield a solution to the original problem. There are 3 basic ways to represent a graph in memory (objects and pointers, matrix, and adjacency list); familiarize yourself with each representation and its pros/cons. You should know the basic graph traversal algorithms, breadth-first search and depth-first search. Know their computational complexity, their tradeoffs and how to implement them in real code.

Recursion: Many coding problems involve thinking recursively and potentially needing a recursive solution. Prepare for recursion, which can sometimes be tricky if not approached properly. Practice some problems that can be solved iteratively, but a more elegant solution is recursion.

Operating systems: You should understand processes, threads, concurrency issues, locks, mutexes, semaphores, monitors and how they all work. Understand deadlock, livelock and how to avoid them. Know what resources a process needs and a thread needs. Understand how context switching works, how it's initiated by the operating system and underlying hardware. Know a little about scheduling. The world is rapidly moving towards multi-core, so know the fundamentals of "modern" concurrency constructs.

Mathematics: Some interviewers ask basic discrete math questions. This is more prevalent at Google than at other companies because counting problems, probability problems and other Discrete Math 101 situations surrounds us. Spend some time before the interview refreshing your memory on (or teaching yourself) the essentials of elementary probability theory and combinatorics. You should be familiar with n-choose-k problems and their ilk - the more the better.



Still want more info?

[Tech interviews @ Google](#)

[Distributed systems & parallel programming](#)

[Scalable Web Architecture & Distributed systems](#)

[How search works](#)

Organizzazione

❖ Il corso è organizzato in

➤ Lezioni ed esercitazioni

- Squadra unica (intero corso)
- 1 blocco da 1.5 ore; Lunedì 13.00-14.30, **Aula R3B**
- 1 blocco da 1.5 ore; Venerdì 13.00-14.30, **Aula 2P**
- 1 blocco da 1.5 ore; Venerdì 14.30-16.00, **Aula 2P**

Recupero

➤ Non vi è alcuna distinzione formale tra lezioni e esercitazioni

- Vengono svolti in aula tutti gli argomenti precedentemente indicati
- Per ciascun argomento, le esercitazioni sono svolte contestualmente alla trattazione della parte teorica

Organizzazione

- Circa 15 ore di esercitazioni assistite in laboratorio
 - 3 squadre
 - 3 blocchi da 1.5 ore, 1 per ciascuna squadra
 - Squadra A: A-CAF, Martedì 13.00-14.30, **Aula 6I**
 - Squadra B: CAG-DA, Martedì 14.30-16.00, **Aula 6I**
 - Squadra C: DB-G, Martedì 16.00-17.30, **Aula 6I**
- L'attività di laboratorio comprende
 - Sistema operativo Linux
 - Comandi di shell e scrittura di script
 - Programmazione concorrente in linguaggio C
 - Esercitazioni su tutti gli argomenti trattati in aula

Aula non Lab

Organizzazione

❖ Riassumendo

- Lezioni ed esercitazioni svolte in presenza
 - Video-registrate (virtual-classroom)
 - Diffuse real-time (senza interazioni da remoto)
- Laboratori in presenza
 - Non video-registrati
- Pianificazione completa prevista (argomenti & co.) disponibile su Dropbox

3 lezione
0 lab

3 lezioni
0 lab

3 lezioni
0/1 lab ?

2 lezioni ; **1** laboratorio
(tranne recuperi e casi particolari)

Settimana 1

Settimana 2

Settimana 3

...

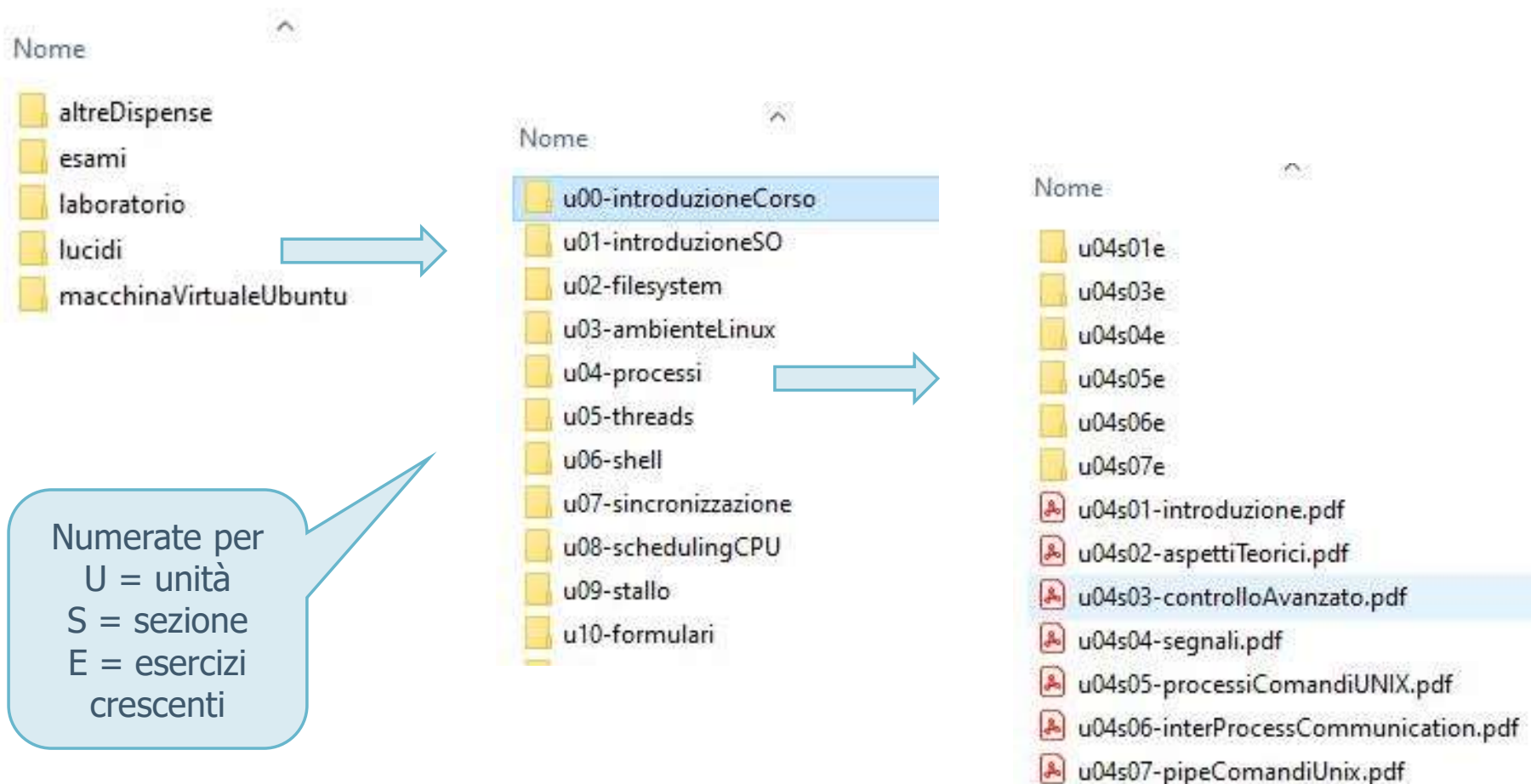
Settimana 14

Materiale disponibile

- ❖ Informazioni generali (portale didattica)
 - Calendario anno accademico, regole e scadenze
 - Prenotazione e risultati esami
 - Archivio storico video-lezioni
 - Il corso è video-registrato da diversi anni

Materiale disponibile

- ❖ Materiale utilizzato (**dropbox** portale didattica)
 - trasparenze proiettate in aula



Materiale disponibile

❖ Materiale utilizzato (**dropbox** portale didattica)

- trasparenze proiettate in aula
- Materiale utilizzato in laboratorio
 - 1 esercitazione alla settimana
 - 1 soluzione alla settimana (settimana precedente)
- Prove d'esame
 - Testi (dall'A.A. 2019-2020 in poi)
 - Soluzioni (idem)
- Selezione dispense e opuscoli
 - Argomenti specifici (wikipedia, stack overflow, etc.)
- Macchina virtuale Ubuntu-24.04 LTS pre-installata

2012-2013 → 2018-2019
non completamente
compatibili

Testi, letture e dispense

❖ Testi consigliati

➤ Aspetti teorici

- A. Silberschatz, P. B. Galvin, G. Gagne, " Sistemi Operativi. Concetti ed esempi", decima edizione, Pearson Education Italia, 2019, ISBN: 978-88-9190-455-3

Capitoli 1-8, 13, 14 (10 capitoli su 19)



Testi, letture e dispense

❖ Testi consigliati

➤ Aspetti teorici

- Andrew S. Tanenbaum, Herbert Bos, "I moderni sistemi operativi", quinta edizione, Pearson Education Italia, 2023, ISBN: 978-88-9193-195-5

Alternativa
(+ conciso e tecnico,
diversi capitoli on-line)



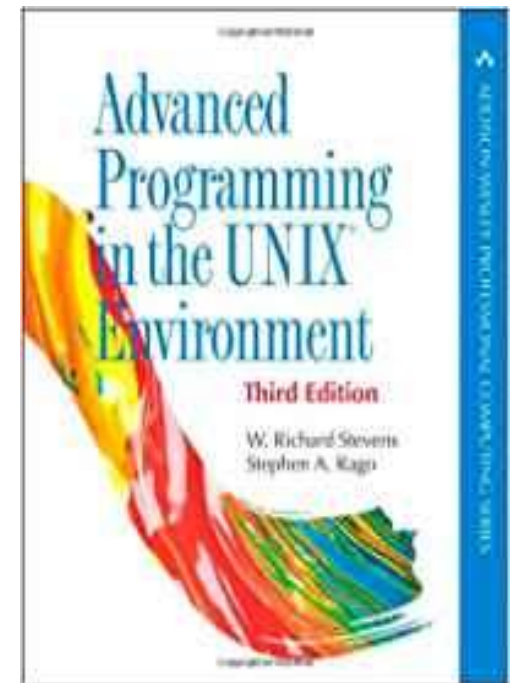
Testi, letture e dispense

❖ Testi consigliati

➤ Ambiente UNIX/Linux

- W. R. Stevens, S. A. Rago, "Advanced programming in the UNIX Environment", terza edizione, Addison-Wesley, 2013, ISBN: 978-0-321-63773-4

Guida sistemi UNIX



Valutazione e regole d'esame

- ❖ Il calendario e le regole sono specificate dal “Manifesto agli Studi” dell’anno accademico
 - Le date degli appelli sono definite da un’apposita **commissione** e **non** dal docente
- ❖ Non esistono prove intermedie, prove di esonero oppure valutazioni di laboratorio
- ❖ È indispensabile
 - Prenotarsi per ogni appello che si intende sostenere
 - Cancellare la propria prenotazione in caso non si intenda presentarsi

Indispensabile per poter utilizzare la piattaforma “Esami”

Erano in presenza
"su carta"

Valutazione e regole d'esame

❖ La prova d'esame consiste in un'unica prova **scritta** della durata di **130** minuti

Erano 100 in
presenza "su carta"

➤ Le prove in presenza

- Verranno effettuate in aula (o laboratorio) utilizzando la piattaforma **Esami** sotto l'assistenza dei docenti
- Procurarsi e verificare gli strumenti necessari con dovuto anticipo
 - Hardware (connessione wi-fi, standard prese, etc.)
 - Software (app, aggiornamenti, etc.)

Verifica/Utilizzo
piattaforma Moodle

➤ Non esiste una prova orale né la possibilità di sostenere una prova orale

Valutazione e regole d'esame

➤ Durante la prova

- Inserire spiegazioni nel testo della risposta, in caso di dubbi (o nel testo di una domanda "aperta")
- È possibile utilizzare
 - Fogli bianchi con penna/matita (calcoli, etc.)
 - Le **3** mini-dispense ("formulari omologati") fornite dal docente (comandi UNIX, thread, BASH)
 - Esse saranno **inserite direttamente** come link all'interno della domanda che specifica la possibilità di ritirarsi
 - Calcolatrice
 - **Inserita direttamente** dai docenti nelle domande che la richiedono
- **Non è ammesso l'utilizzo di**
 - Appunti, libri, calcolatrici, cellulari, etc.
 - I cellulari e gli altri dispositivi elettronici vanno **spenti**

Valutazione e regole d'esame

- Le domande prevedono la conoscenza di
 - Aspetti teorici: Verifica delle **conoscenze** acquisite
 - Aspetti pratici: Verifica delle **competenze** acquisite tramite problem-solving
 - Esercizi di C e BASH, programmi, comandi, script

- La prova comprende **6-18** domande

Erano 6
(nei compiti su carta)

- A risposta aperta
 - Corrette dai docenti
- A risposta chiusa, multipla, vettoriale, numerica, etc.
 - Corrette **automaticamente** dalla piattaforma e per le quali è indispensabile rispettare il formato

Possono avere
punteggi negativi

Valutazione e regole d'esame

- Ogni risposta viene valutata da **1** a **6** punti
 - Il punteggio finale è dato alla somma dei punti acquisiti in ciascun esercizio, con un massimo di **36**
 - L'arrotondamento è per eccesso
 - Ogni voto $\geq x.5$ è arrotondato a $x+1$
 - $25.49 \rightarrow 25$, $25.50 \rightarrow 26$
 - Ogni voto ≥ 17.25 (and ≤ 18) è arrotondato a 18
 - $17.24 \rightarrow \text{Fail}$, $17.25 \rightarrow 18$
 - Voti $\geq \mathbf{32}$ (o **33** a seconda della difficoltà del compito) sono convertiti in **30 e lode**

Estratto

Risultati anni accademici precedenti

❖ Risultati esami per coorte

➤ Dal 2012-2013 al 2022-2023

➤ Valutati al 01.10.2023

11 coorti

I = Iscritti
S = Superi
P= Percentuale

Anno Accademico	2018-2019			2019-2020			2020-2021		
	I	S	P[%]	I	S	P[%]	I	S	P[%]
2018-2019	231	153	66						
2019-2020	69	45	20	175	122	70			
2020-2021	22	12	5	48	28	16	192	116	61
2021-2022	9	3	1	25	3	2	58	30	26
2022-2023	4	0	1	7	0	0	21	8	4
Superi totali		213	92		153	87		154	80
Ritiri		14	6		15	9		25	13
In coda		4	2		7	4		13	7

Risultati anni accademici precedenti

❖ Risultati esami complessivi

- Dal 2012-2013 al 2022-2023
- Valutati al 01.10.2023

Numero totale di ...	Totale	[%]
... studenti iscritti	2009	100.0%
... registrazioni (esami sostenuti)	2596	129.2%
... studenti mai presentatisi	297	14.8%
... ritiri dal corso (sino al 2020-2021)	253	12.6%
... superi (sui totali = 1626/2009)	1626	80.9%
... superi (sui presentatisi = 1626/(2009-297))		95.0%
Voto medio	24.3	

Risultati ultimo anno accademico

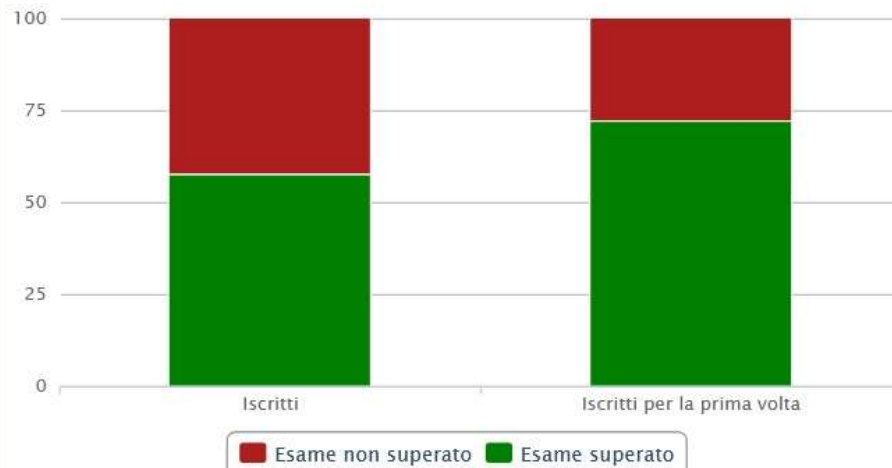
Appello di
settembre NON
contabilizzato

2023-2024
106 nuovi, 191 totali

2024-2025
104 nuovi, 180 totali

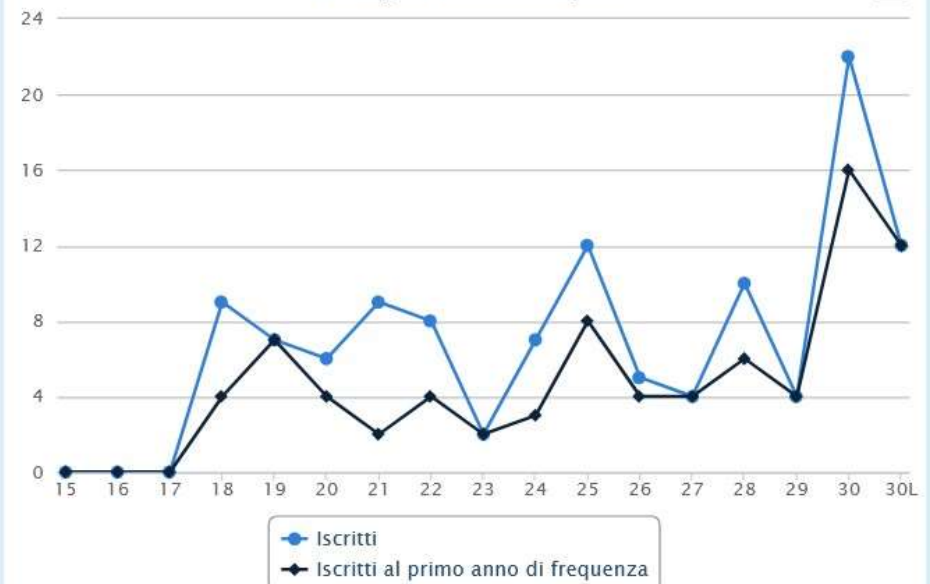
05CJCOA – Sistemi operativi
QUER STEFANO
A.A. 2023/24

Totale iscritti: 202 (111) – Superi: 117 (80 al primo anno)



Highcharts.com

Dettaglio voti dei superi



Highcharts.com

Commenti (critiche) pregressi

❖ Organizzazione

➤ Corso troppo compresso come numero di ore

- Aumentare numero di crediti (da 6 a 8 o 10)
- Tagliare altri argomenti e/o corsi non informatici

Hug?

Ci si muove lentamente in questa direzione

Eliminati alcuni argomenti (u03s05, u04s02 in parte, u06s03, un laboratorio, etc.)

- Approfondire meglio i vari argomenti
- Utilizzare aule migliori (e.g., con prese)

Hug?

Commenti (critiche) pregressi

❖ Contenuti

- Il corso è chiamato "sistemi operativi" ma
 - Si danno nozioni generali solo di sistemi Unix-Like
 - Sembra la continuazione del corso di APA

Vero

Seguito: "Programmazione di sistema"
("System and Device Programming")

- Ridurre la parte di concorrenza e di BASH, trattare networking, socket, e aumentare il legame con calcolatori elettronici

Hug?

Funny “Linus Torvalds” Quotes

Nel 1991 inizia a sviluppare il kernel Linux



- ❖ My name is Linus, and I am your God
- ❖ In my opinion MS is a lot better at making money than it is at making good operating systems
- ❖ A computer is like air conditioning. It becomes useless when you open Windows
- ❖ Microsoft isn't evil, they just make really crappy operating systems
- ❖ I am not out to destroy Microsoft, that would be a completely unintended side effect

Riassumendo ...

- ❖ Nel corso affronteremo argomenti teorici (lezioni in aula) e pratici (lezioni in aula e esercitazioni in laboratorio)

Riassumendo ...

- ❖ Nel corso affronteremo argomenti teorici (lezioni in aula) e pratici (lezioni in aula e esercitazioni in laboratorio)
 - La teoria è quando si sa tutto, ma non funziona nulla

Riassumendo ...

- ❖ Nel corso affronteremo argomenti teorici (lezioni in aula) e pratici (lezioni in aula e esercitazioni in laboratorio)
 - La teoria è quando si sa tutto, ma non funziona nulla
 - La pratica è quando funziona tutto ma non si sa il perché

Riassumendo ...

- ❖ Nel corso affronteremo argomenti teorici (lezioni in aula) e pratici (lezioni in aula e esercitazioni in laboratorio)
 - La teoria è quando si sa tutto, ma non funziona nulla
 - La pratica è quando funziona tutto ma non si sa il perché
 - **In questo corso, teoria e pratica saranno combinate: non funzionerà nulla e non si saprà il perché**

(forse) Albert Einstein, 1879-1955