

```
#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)
    {
        fprintf(stderr, "ERRORE: serve un parametro con il nome del file\n");
        exit(1);
    }
    f = fopen(argv[1], "r");
    if(f==NULL)
    {
        fprintf(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

Informazioni generali

- ❖ Sistemi Operativi
 - 05CJC_{OA}
 - ING-INF/05
- ❖ Laurea in Ingegneria Informatica (INF1T3)
- ❖ Anno 3, semestre 1
- ❖ 6 crediti, 60 ore
- ❖ Primo corso
 - Studenti da AA a GZ (+debito esami)

Docenza e assistenti



Quer Stefano, DAUIN
011-090-7076
stefano.quer@polito.it
<http://fmgroup.polito.it/quer/>

Palena Marco, DAUIN
011-090-7048
marco.palena@polito.it



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

Presentazione

- ❖ Il modulo ha lo scopo di
 - Introdurre gli elementi di base dell'architettura dei sistemi operativi
 - Approfondire i concetti e le tecniche di programmazione concorrente
 - Fornire conoscenze specifiche dei sistemi operativi UNIX-like, con particolare riferimento ai sistemi Linux

Risultati attesi

- ❖ Il corso fornisce conoscenze su
 - Caratteristiche, architettura e funzionalità di base dei principali sistemi operativi moderni
 - Amministrazione di sistemi UNIX/Linux mediante utilizzo di comandi, shell, filtri e linguaggi di script
 - Gestione di processi e di thread
 - Utilizzo della programmazione concorrente per la risoluzione di problemi elementari
 - Problemi classici di sincronizzazione

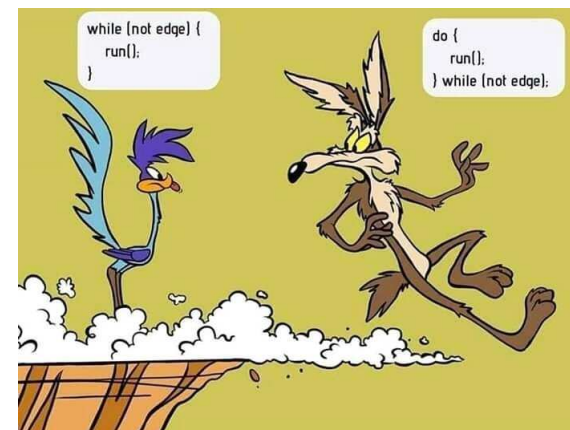
Prerequisiti

❖ 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

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

Contenuto

❖ 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](#)

Contenuto

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 finding 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
 - Esercitazioni
 - Intero corso (unica squadra)
 - 2 blocchi da 1.5 ore (giovedì)
 - 1 blocco da 1.5 ore (martedì; occasionale)
- ❖ Non vi è alcuna distinzione formale tra lezioni e esercitazioni
 - Le esercitazioni, relative ai vari argomenti, sono svolte contestualmente alla trattazione della parte teorica

Organizzazione

- ❖ Lezioni e esercitazioni sono estese da circa 15 ore di esercitazioni assistite in laboratorio
 - 3 squadre
 - 3 blocchi da 1.5 ore (1 per ciascuna squadra)
 - Squadra A: A – CAS, Martedì 16.00-17.30, laib 3
 - Squadra B: CAT – DI, Giovedì 17.30-19.00, laib 3
 - Squadra C: DJ – GZ, Venerdì 10.00-11.30, laib 1
- ❖ 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

Testi, letture e dispense

❖ Materiale

➤ Pagina personale studente (portale Politecnico)

- Video-lezioni (2017-2018)
- Calendario anno accademico, regole e scadenze
- Prenotazione e risultati esami

➤ Pagina WEB docente

```
http://staff.polito.it/stefano.quer/  
http://fmgroup.polito.it/quer/
```

- Materiale utilizzato durante il corso
 - Lucidi, dispense e materiale vario
 - Esercizi e soluzioni per il laboratorio
 - Testi e soluzioni prove d'esame

Testi, letture e dispense

❖ Materiale utilizzato dal docente

- trasparenze proiettate in aula
- Materiale utilizzato in laboratorio
 - Esercitazioni di laboratorio (1 per settimana)
 - Soluzioni esercitazioni di laboratorio (1 per settimana con 1 settimana di ritardo)
- Dispense e opuscoli su argomenti specifici
 - Ricerche su google o simili
 - Siti specifici (wikipedia, stack overflow, etc.)
 - Selezione dispense su sito docente

Numerate per
U = unità
S = sezione
E = esercizi
crescenti

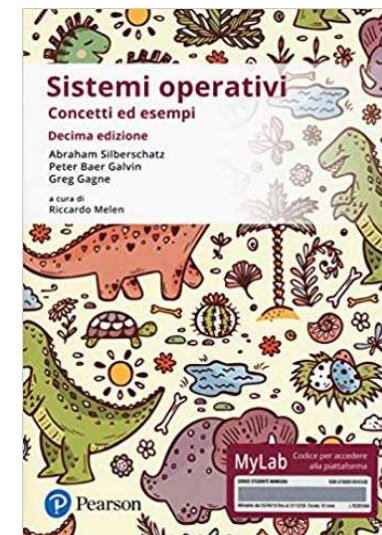
Testi, letture e dispense

❖ 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, "I moderni sistemi operativi", quarta edizione, Pearson Education Italia, 2019, ISBN: 978-88-9190-101-9

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



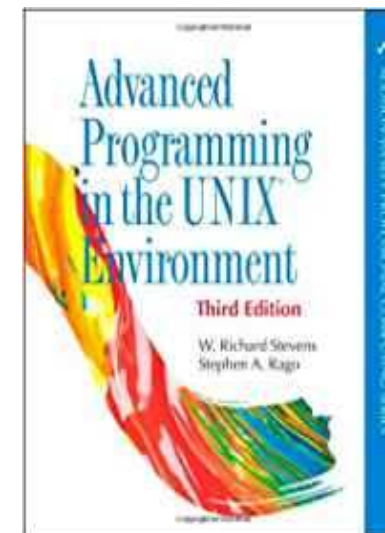
Testi, letture e dispense

❖ 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 corrente
 - 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

Valutazione e regole d'esame

- ❖ La prova d'esame consiste in un'unica prova **scritta** della durata di **100** minuti
 - Non esiste una prova orale né la possibilità di sostenere una prova orale
 - Durante la prova
 - È possibile ritirarsi in qualsiasi momento
 - È ammesso l'utilizzo di **3** mini-dispense ("formulari omologati") fornite dal docente (comandi UNIX, thread, bash) in originale (non manoscritte)
 - **Non è ammesso alcun altro materiale** (no appunti, no libri di testo, no calcolatrici, etc.)
 - I cellulari vanno **spenti** e **appoggiati** sul banco a fianco del candidato, durante l'intera prova scritta

Valutazione e regole d'esame

- La prova comprende **6** domande
- Le domande includono
 - La verifica delle conoscenze acquisite, tramite domande su aspetti teorici
 - Il controllo della capacità acquisite, tramite esercizi di C e UNIX/Linux (programmi, comandi e script)
- Ciascun esercizio ha un peso uguale a **6** punti
 - Il punteggio finale è pari alla somma dei punti acquisiti in ciascun esercizio
 - $6 \times 6 = 36$, con arrotondamento per eccesso
 - Voti $\geq 32/33$ sono automaticamente convertiti in 30 e lode

Da febbraio 2020 la domanda di **AWK** verrà eliminata

Eliminato
2012-2013

Risultati anni accademici precedenti

❖ Risultati esami per coorte

- Dal 2012-2013 al 2018-2019
- Valutati al 30.09.2019

I = Iscritti
S = Superi
P= Percentuale

Anno Accademico	2013-2014			2014-2015			2015-2016		
	I	S	P[%]	I	S	P[%]	I	S	P[%]
2013-2014	210	146	70						
2014-2015	56	27	13	225	142	63			
2015-2016	10	3	1	41	24	11	170	109	64
2016-2017	6	0	0	15	6	3	52	18	11
2017-2018	3	1	1	8	2	1	23	6	4
2018-2019	2	1	1	6	2	1	16	4	2
Superi totali		178	85		176	78		137	81
Ritiri		31	15		45	20		21	12
In coda		1	1		2	2		12	7

Eliminato
2012-2013

Risultati anni accademici precedenti

❖ Risultati esami per coorte

- Dal 2012-2013 al 2018-2019
- Valutati al 30.09.2019

I = Iscritti
S = Superi
P= Percentuale

Anno Accademico	2016-2017			2017-2018			2018-2019		
	I	S	P[%]	I	S	P[%]	I	S	P[%]
2013-2014									
2014-2015									
2015-2016									
2016-2017	198	147	74						
2017-2018	40	22	11	219	155	71			
2018-2019	14	3	2	58	29	13	231	153	66
Superi totali		172	87		184	84		153	66
Ritiri		6	3		6	3		0	0
In coda		29	13		29	13		78	34

Risultati anni accademici precedenti

❖ Risultati esami complessivi

- Dal 2012-2013 al 2018-2019
- Valutati al 30.09.2019

Numero totale di ...	Totale	[%]
... studenti iscritti	1410	100.0%
... registrazioni (esami sostenuti)	1879	133.3%
... studenti mai presentatisi	177	12.6%
... ritiri dal corso (sino al 2018-2019)	136	9.6%
... superi (sui totali = 1145/1410)	1145	81.2%
... superi (sui presentatisi = 1145/(1410-177))	953	92.9%
Voto medio	24	

Commenti (critiche) studenti 2018-2019

❖ Carico

- Corso troppo compresso come numero di ore
 - Aumentare numero di crediti
 - Tagliare altri argomenti/corsi
 - Approfondire meglio i vari argomenti

❖ Argomenti

- Il corso è chiamato "sistemi operativi" ma
 - Si danno nozioni generali solo di sistemi Unix-Like
 - Sembra la continuazione del corso di APA
- Eliminare BASH
 - Fare Python (o Ruby)

Commenti (critiche) studenti 2017-2018

❖ Organizzazione

- Aule e orari scomodi
 - "Rimuovere dall'orario le ore non utilizzate"
- Lucidi con molti errori
 - "Codice scritto a casa"
- Gestione laboratori critica
- Tempo valutazione appelli

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