

Introduzione a SETI

... regole e “burocrazia”

Giovanni Chiola Giovanni Lagorio

SETI a.a. 2022/23

Iscrivetevi al corso **SISTEMI DI ELABORAZIONE E TRASMISSIONE DELL'INFORMAZIONE - 80300** su Aulaweb

<https://2022.aulaweb.unige.it/course/view.php?id=310>

Obbligatorio

Tutte le comunicazioni ufficiali, i testi/consegne delle esercitazioni, . . . , etc passano da lì

Forum:

- Annunci
 - iscrizione obbligatoria
- Discussioni generali
 - iscrizione facoltativa, ma consigliato

Due docenti:

- Giovanni Chiola
 - `chiolag@acm.org`
- Giovanni Lagorio
 - `giovanni.lagorio@unige.it`

Domande

Nel 99.99% dei casi, invece della mail ai docenti è meglio:

- fare domande a lezione
- usare il forum “Discussioni generali” su AulaWeb

così serve a tutti

Domande generali

Non risponderemo singolarmente per dubbi generali; mettiamo le cose in prospettiva:

- Ci sono quasi 300 persone su AulaWeb per SETI (dato del 2020/21)
 - non avrebbe senso rispondere a tantissime domande molto simili
 - soprattutto, rischieremmo di dare risposte (lievemente) diverse a persone diverse
- Per dubbi/domande, per esempio sul testo di un'esercitazione, **è giusto che tutti ricevano la stessa identica risposta** (e sul forum rimane per chi, magari, affronterà la problematica in seguito)

Se, invece, parliamo di un problema specifico della vostra implementazione, non avrebbe senso condividere sul forum il vostro codice con tutti gli altri. In quel caso, potete chiedere direttamente via email. Però ...

Domande tecniche

Non ha senso scrivere genericamente “non va boh; ecco tutti i miei sorgenti, mi può dire dove sbaglio?”

Dovrete comportarvi come se stesse preparando un *bug report*, per cui **dovete specificare:**

- dove si verifica il problema
- **i passi per riprodurre il problema**
- qual è l'output atteso e quale, invece, è quello che ottenete
- **perché ritenete che l'output dovrebbe essere diverso**; quale pagina di manuale (o altra fonte di documentazione) avete consultato

In generale, tutte le chiamate precedenti al punto dove ottenete il comportamento strano sono andate a buon fine?

Molto spesso i problemi che segnalate sono dovuti al fatto che ignorate i codici di errori (cosa che vi abbiamo ripetutamente detto di non fare 😊)

Salvo novità, in *aula 506* e laboratorio *Software 1*

- Lunedì 16-18 @ 506/SW1
- Martedì 9-11 @ 506
- Mercoledì 14-16 @ 506
- Venerdì 16-18 @ 506/SW1

All'inizio, solo in aula

Anticiperemo il “quarto d'ora accademico” al martedì e mercoledì, in modo da iniziare alle 9:30/14:30

DaD: no, grazie

Quest'anno **la didattica è in presenza**

Crediamo che le registrazioni delle lezioni siano utili, per cui saranno *sperimentalmente* **registrate in modalità “best effort”**; *non* sarà DaD:

- non è garantita la registrazione di tutte le lezioni (e potremmo interrompere l'esperimento in qualsiasi momento)
- non sarà possibile interagire da remoto
- i video non saranno necessariamente disponibili subito dopo la lezione

Argomenti e prerequisiti

SET(I) mette insieme argomenti di:

- sistemi operativi (\Rightarrow Giovanni Lagorio) e
- reti di calcolatori (\Rightarrow Giovanni Chiola)

Costruendo sui contenuti di IP/ASD/AdC:

- programmare in C(++)
 - (de)allocazione della memoria e puntatori come se piovevano ☺
- architettura di un calcolatore convenzionale

Lecture consigliate

- “Everything you need to know about pointers in C”
<https://boredzo.org/pointers/>
- “Modern C” by Jens Gustedt
<http://modernc.gforge.inria.fr/>

- I vostri appunti
- I video (se/quando disponibili)
- Per alcuni argomenti, i lucidi, ma, da soli, *non saranno sufficienti*
- Le pagine del manuale — sono **FONDAMENTALI**
- Aulaweb
 - Link di approfondimento/materiale non coperto dai libri “ufficiali”
 - Forum
- Codice!
Talk is cheap. Show me the code.
Linus Torvalds
per esempio, Linux, Xv 6, ... (ne parleremo)
- Libri...

Seguiti (abbastanza/con integrazioni):

- **Operating Systems: Three Easy Pieces**, di Arpaci-Dusseau;
<http://pages.cs.wisc.edu/~remzi/OSTEP/>
- **The Linux Command Line**, di William Shotts;
<http://linuxcommand.org/tlcl.php>

Libri “classici”, con traduzioni in Italiano

- Modern Operating Systems, di A. S. Tanenbaum e H. Bos
- Operating System Concepts, di Silberschatz, Galvin e Gagne

Per approfondimenti:

- What Every Superuser Should Know di B. Ward
- The Linux Programming Interface: A Linux and UNIX System Programming Handbook, di M. Kerrisk

- **Computer Networking, a Top-Down Approach** di James F. Kurose e Keith W. Ross, di cui esiste anche la traduzione in Italiano

Carico di lavoro

Corso da x crediti = $x \cdot 25$ ore di lavoro dello studente *medio* per un risultato *buono*. Per SETI $x = 12$ (erano 15 per SET), quindi **300 (375) ore di lavoro**

- Medio = non un genio, né uno stupido ma con il background necessario, che segue il processo di apprendimento suggerito e sfrutta tutte le risorse offerte
- Per risultati ottimi uno studente non eccellente può dover impiegare più sforzo

Si cerca di migliorare, basandosi sull'anno precedente e il vostro feedback:

- corso troppo pesante incide sul rendimento globale (interferenze con gli altri corsi)
- corso troppo leggero vi froda dei vostri diritti (pagate per imparare)

12 (15) CFU, 300 (375) ore di lavoro

Gli argomenti sono “tanti”

- si tratta di un corso impegnativo
- incontri 4 giorni su 5 (era: *tutti* i giorni)

Vi occuperà parecchie (circa 200/250) ore di studio/lavoro individuale, circa:

- metà durante il semestre, per rispettare le scadenze per le consegne delle esercitazioni (ne parliamo fra un attimo), e
- altrettante per rivedere/approfondire la teoria prima dell'esame

Consiglio spassionato

Cercate di mantenere il passo durante il semestre, altrimenti NON vi basteranno 200/250 ore dopo

SETI (12 CFU) vs SET (15 CFU)

I seguenti argomenti:

- dialogo con le periferiche (polling, interrupt e DMA)
- virtualizzazione della memoria: segmentazione, paginazione (MMU/TLB)
- modalità utente/supervisore della CPU e introduzione a kernel e processi

sono “transitati” da SET ad AdC.

A SETI riprendiamo/approfondiamo alcune cose; quindi, sostanzialmente, i primi argomenti (di s.o.) che venivano trattati in SET, sono adesso svolti in AdC e quindi diventati pre-requisiti per SETI

Esercitazioni (1/2)

- Tre esercitazioni: *ping-pong*, *μbash* e *incApache*
- Presentate a lezione, ma **testo/documentazione/consegna su AW**
- Vi servono per vedere in pratica i concetti illustrati a lezione; utile strumento per controllare se state davvero capendo
 - la **conoscenza di concetti, syscall, funzioni, comandi, ...** necessari per svolgere le esercitazioni **fa parte del programma d'esame**
 - le esercitazioni (e il fatto che le abbiate capite/siano farina del vostro sacco) hanno un peso significativo nel determinare il voto finale
 - **“darci una botta”**, senza approfondire i concetti, è un classico **autogol**
 - è importante **testare** la vostra implementazione, sia su **input “buoni”** (esercitando almeno una volta *tutto* il flusso di esecuzione) **e su quelli “cattivi a piacere”**

Esercitazioni (2/2)

- Si lavora in **gruppi di due/tre persone (autogestiti)**, *ma*
- Le **consegne sono individuali**, ciascun membro del gruppo
 - consegna la stessa soluzione, precisando la composizione del gruppo
 - deve aver capito, saper discutere e/o apportare piccole modifiche su *tutto* il codice consegnato
- La collaborazione è incentivata, il *copia-incolla-senza-aver-capito-una-mazza* no
- Solo chi consegna tutte le esercitazioni in tempo può aspirare alla lode
- **Le esercitazioni di un a.a. vanno consegnate e discusse durante quello stesso a.a.**
- **Quelle discusse con esito positivo valgono anche per gli a.a. successivi, mentre tutte le altre non valgono più nulla dopo la fine dell'a.a. corrispondente**

- Saper implementare correttamente una specifica è un requisito essenziale per qualsiasi informatico, al di là delle esercitazioni di SETI
- Consegne in cui non funzionano nemmeno esempi minimali, inclusi tutti gli esempi del testo, saranno considerate insufficienti
 - Quindi, dovranno essere corrette e riconsegnate in un appello successivo
- Siccome noi testeremo le vostre consegne su Linux
 - abbiamo riscontrato alcuni comportamenti particolari sui Mac
 - sviluppate/testate su una macchina (eventualmente virtuale) Linux
 - su Windows, WSL 2 (non WSL 1) dovrebbe essere ok (perché, alla fin fine, è una virtual-machine Linux)

Esami: quiz ed esercitazioni

- Cinque appelli, che iniziano con un test/“quiz”
 - Ci saranno risposte (molto sbagliate) con valutazione *negativa*
 - Se non sapete una risposta, conviene non rispondere che tirare a caso
 - Si svolgeranno in presenza
- Salvo aver già discusso con esito positivo le esercitazioni
 - Prima di affrontare il test bisogna aver (ri)consegnato tutte le esercitazioni dell'a.a. corrente
 - In caso contrario, il test non viene corretto/considerato
 - La prima volta che si consegna un test, indipendentemente dal suo risultato, si discuteranno le esercitazioni consegnate
- I test sono una **valutazione preliminare** per l'ammissione all'orale
 - Non hanno un voto, il risultato è ammesso/non-ammesso all'orale
 - Un test vale fino alla data del test successivo (o il termine dell'a.a.)
- Dopo ogni test prepareremo il calendario delle discussioni/orali
 - Nel test avrete modo di specificare eventuali vincoli

Esami: discussione esercitazioni e orale

Nella discussione sulle esercitazioni verificheremo che sappiate cosa fa (*tutto*) il codice consegnato e che sia “farina del vostro sacco”. Potremmo chiedervi di spiegarne il funzionamento e/o apportare piccole modifiche

- Se possibile, portate un vostro PC con le esercitazioni svolte

L'esame si conclude con una prova orale, che si può sostenere solo se

- il risultato del test e
- la discussione delle esercitazioni

hanno avuto un esito positivo.

L'esame orale include sia la parte di teoria, sia il contenuto/gli argomenti delle esercitazioni (che fanno parte del programma d'esame anche se discusse precedentemente).

That's all, folks!

Domande?