

# Corso di Reti di Calcolatori e Laboratorio

## Progetto Esame

Docenti: Prof. Emanuel Di Nardo

A. A. 2023/2024

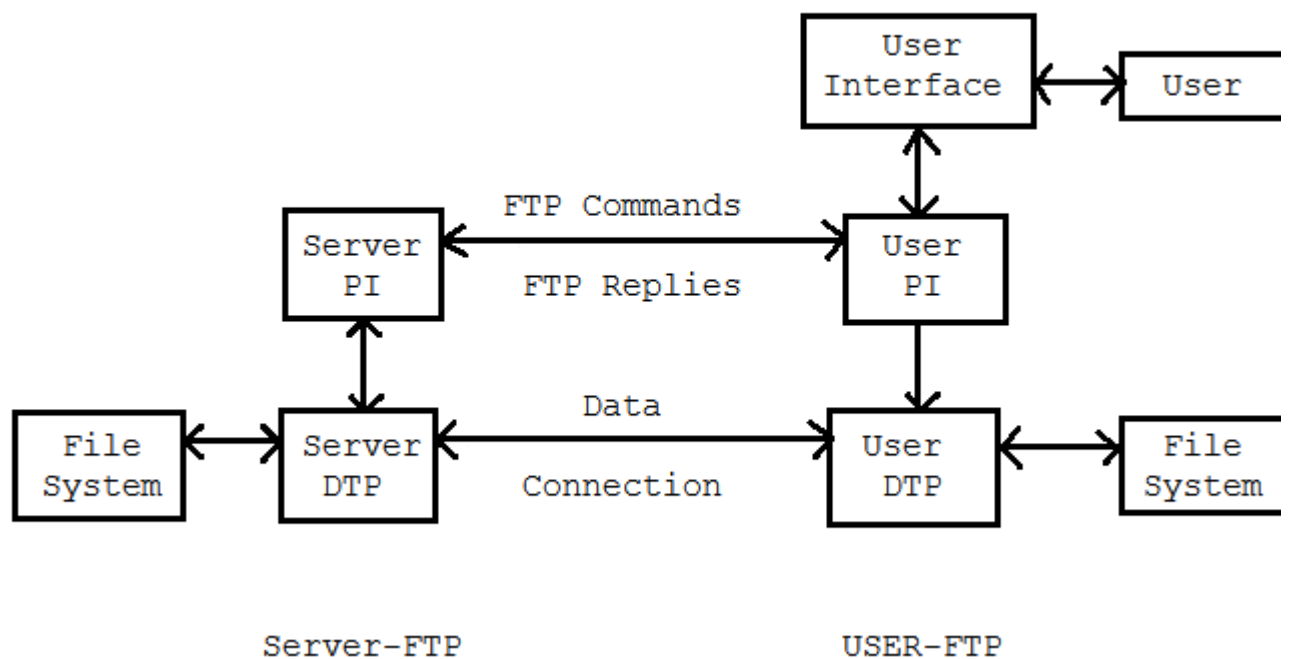
### Studente

- **Cognome:** Angarella
- **Nome:** Ciro
- **Matricola:** 0124002559
- **Codice gruppo:** btlu25c203
- **Componenti gruppo:** 3

### Traccia - FTP

Implementare un servizio di scambio dati seguendo le specifiche del protocollo FTP

(Estratto da wikipedia)



- **PI** (*protocol interpreter*) è l'interprete del protocollo, utilizzato da client (User-PI) e server (Server-PI) per lo scambio di comandi e risposte. In gergo comune ci si riferisce a esso come "canale comandi".

- **DTP** (*data transfer process*) è il processo di trasferimento dati, utilizzato da client (User-DTP) e server (Server-DTP) per lo scambio di dati. In gergo comune ci si riferisce a esso come "canale dati".

Utilizzare i comandi FTP definiti in RFC 959 per il canale **PI**. Stabilire una connessione **Passiva**.

Il sistema deve permettere il login ed il logout di due tipologie di utenti:

- Utente registrato
  - L'utente può vedere la sola directory a lui assegnata
  - Può effettuare download/upload
  - Può rinominare e rimuovere i file
  - Può avere la lista dei file
- Utente anonimo
  - Può vedere solo una directory condivisa per utenti anonimi
  - Può effettuare solo il download
  - Può avere la lista dei file

Utilizzare un file o una struttura dati hardcoded per gli utenti registrati

Il sistema **non** deve contenere:

- Creazione di una directory
- Navigazione nel filesystem (limitato alle cartelle condivise del progetto)
  - Es. non è possibile navigare nella home dell'utente
- Creazione directory
- Rinomina e rimozione directory
- Sistema di registrazione
- Protocolli di sicurezza

**N.B.** Non usare le porte riservate utilizzate solitamente da FTP

## Invio di file

Per inviare file utilizzare la funzione sendfile nella libreria sys/sendfile.h

Riferimento: <https://man7.org/linux/man-pages/man2/sendfile.2.html>

## Note di sviluppo

La prova d'esame richiede la progettazione e lo sviluppo della traccia proposta.

Il progetto deve essere sviluppato secondo le seguenti linee:

- utilizzare un linguaggio di programmazione a scelta (C, Java, Python, etc...)
- utilizzare una piattaforma **Unix-like**;
- utilizzare le **socket**;
- inserire sufficienti **commenti**;

## Consegna progetto

# Documentazione

Lo studente deve presentare la documentazione relativa al progetto. La documentazione deve contenere:

- Descrizione del progetto;
- Descrizione e schema dell'architettura;
- Dettagli implementativi dei client/server;
- Parti rilevanti del **codice** sviluppato;
- Manuale utente con le istruzioni su **compilazione** ed **esecuzione**;

E' possibile redigere la documentazione usando latex o markdown

**Per chi usa latex.** Si consiglia di utilizzare la piattaforma Overleaf:

- <https://www.overleaf.com/>

**Per i markdown.**

- <https://mystmd.org/>
- Pagine descrittive usando Jekyll (<https://jekyllrb.com/>) o Hugo (<https://gohugo.io/>)
  - Consigliato usare le github pages (<https://pages.github.com/>)

## Formato consegna

Ogni gruppo deve consegnare tutti i file e la documentazione tramite un servizio git remoto (**github**, **gitlab**, ...):

- Creare un repository pubblico!
- Ogni partecipante del gruppo deve essere aggiunto come collaboratore
- Dare nomi significativi ai commit

# Consegna

Il progetto va consegnato tramite email al docente

**emanuel.dinardo@uniparthenope.it**

- **Obbligatorio** inviare l'email dall'account studente
- Inserire *Nome, Cognome* e *Marticola* di tutti i membri del gruppo
- Inserire il link al repository github
- **Entro** una settimana dall'esame

## Modalità di esame

L'esame consisterà nella discussione del progetto con possibili domande sulla parte pratica e progettuale e domande di teoria.

I progetti di gruppo devono essere discussi **OBBLIGATORIAMENTE** da tutti i membri lo stesso giorno.

© 2024, theme made with ❤ by **Creative Tim** for a better web.

[Creative Tim](#)[About Us](#)[Blog](#)[License](#)