Universita` degli Studi di Napoli “Parthenope” Facolta` di Scienze e Tecnologie

Corso di laurea in Informatica



Progetto di Reti di Calcolatori

“Universit`a”

DOCENTE STUDENTI

Emanuel Di Nardo Alberto Gambalonga – 0124002583 Fabio Cosentino – 0124002490

Biagio De Blasio – 0124002424

Anno Accademico 2023-2024

**Indice**

[Traccia 3](#_Toc168164615)

[Capitolo 1 - Descrizione del progetto e dell’architettura 4](#_Toc168164616)

[1.1 - Componenti coinvolte 5](#_Toc168164617)

[Capitolo 2 - Manuale utente 7](#_Toc168164618)

[2.1 - Server universitario 8](#_Toc168164619)

[2.1.1 - Compilazione ed esecuzione 8](#_Toc168164620)

[2.2 - Segreteria 9](#_Toc168164621)

[2.2.1 - Compilazione ed esecuzione 9](#_Toc168164622)

[2.3 - Studente 10](#_Toc168164623)

[2.3.1 - Compilazione ed esecuzione 10](#_Toc168164624)

[Capitolo 3 – Casi D’uso 11](#_Toc168164625)

[3.1 – La segreteria inserisce un nuovo appello per un esame 11](#_Toc168164626)

[3.2 – Lo studente chiede alla segreteria le date disponibili di un esame 13](#_Toc168164627)

[3.3 – Lo studente si prenota ad un appello 15](#_Toc168164628)

[3.4 – Lo studente vuole prenotarsi ad un appello inesistente 17](#_Toc168164629)

[3.5 – La segreteria inserisce un appello già esistente 19](#_Toc168164630)

[3.6 – Lo studente tenta di prenotarsi due volte ad uno stesso appello 20](#_Toc168164631)

II

# Traccia

Scrivere un’applicazione client/server parallelo per gestire gli esami universitari.

* **Segreteria**:

– Inserisce gli esami sul server dell’università (salvare in un file o conservare in memoria il dato)

* + Inoltra la richiesta di prenotazione degli studenti al server universitario

– Fornisce allo studente le date degli esami disponibili per l’esame scelto dallo studente

* **Studente**:

– Chiede alla segreteria se ci siano esami disponibili per un corso

– Invia una richiesta di prenotazione di un esame alla segreteria

* Server universitario:
  + Riceve l’aggiunta di nuovi esami
  + Riceve la prenotazione di un esame

Il server universitario ad ogni richiesta di prenotazione invia alla segreteria il numero di prenotazione progressivo assegnato allo studente e la segreteria a sua volta lo inoltra allo studente

Se la segreteria non risponde alla richiesta dello studente questo deve ritentare la connessione per 3 volte. Se le richieste continuano a fallire allora aspetta un tempo random e ritenta. Simulare un timeout della segreteria in modo da arrivare a testare l’attesa random.

# Capitolo 1 - Descrizione del progetto e dell’architettura

Di seguito viene proposto un diagramma per descrivere l’architettura utilizzata nel progetto.

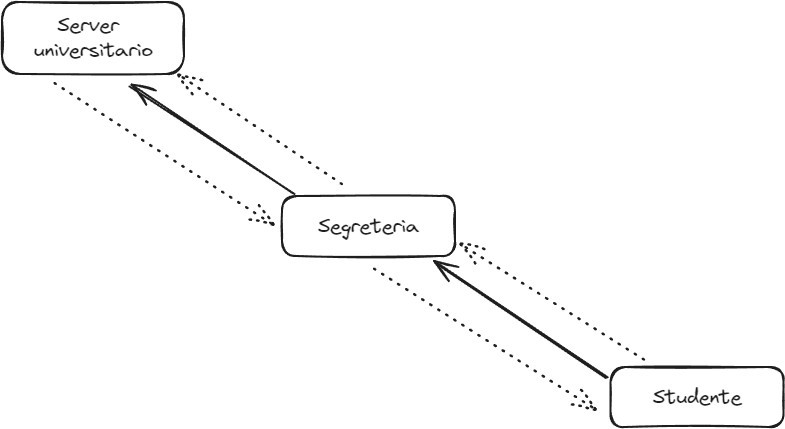


Figura 1.1: Diagramma architettura

Andando a descrivere il diagramma:

* le frecce con linea continua indicano il collegamento tra le componenti;
* le frecce con linea tratteggiata indicano i versi di comunicazione.

*CAPITOLO 1. DESCRIZIONE DEL PROGETTO E DELL’ARCHITETTURA*

## 1.1 - Componenti coinvolte

Da come si evince dalla figura, le componenti coinvolte sono:

* + - **Server universitario**: questa componente funge da **server** per la segreteria e ha i compiti di:
      1. **Aggiungere una prenotazione per un determinato esame**: tale richiesta viene effettuata dallo studente alla segreteria, la quale procede ad inoltrarla al server universitario. Il server universitario memorizza la prenotazione in un file sotto la cartella “resources”.

Inoltre, il server universitario invia alla segreteria il numero di prenotazione progressivo, il quale viene inoltrato allo studente;

* + - 1. **Aggiungere un appello per un determinato esame**: tale richiesta viene effettuata dalla segreteria. Il server universitario memorizza il nuovo appello su un file sotto la cartella “resources”.
    - **Segreteria**: questa componente funge da **server** per lo studente e da

**client** per il server universitario, con compiti di:

#### Inviare al server universitario l’appello di un esame da aggiungere;

* + - 1. **Inoltrare la richiesta di prenotazione ad un appello**, effettuata da uno studente, al server universitario;

#### Fornire allo studente le date degli appelli di un esame specificato.

* + - **Studente**: questa componente funge da **client** per la segreteria, con compiti di:

#### Richiedere alla segreteria gli appelli di un esame specificato;

* + - 1. **Inviare alla segreteria una richiesta di prenotazione di un determinato appello d’esame.**

*CAPITOLO 1. DESCRIZIONE DEL PROGETTO E DELL’ARCHITETTURA*4

Se durante le richieste effettuate, la segreteria smette di rispondere, lo studente ritenta la connessione per tre volte, dopodiché, se continuano a fallire, attende un tempo casuale e ritenta.

# Capitolo 2 - Manuale utente

Di seguito vengono proposte le istruzioni per l’uso dell’applicativo.

Come primo passo, bisogna aprire il progetto posizionandosi all’interno della cartella “**src**“.

*CAPITOLO 2. MANUALE UTENTE*

## 2.1 - Server universitario

Si vanno a descrivere le istruzioni per compilare ed eseguire il codice relativo alla componente del server universitario.

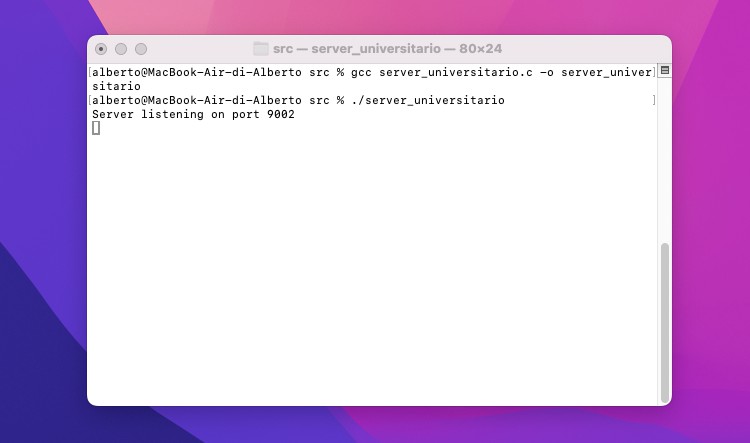
### 2.1.1 - Compilazione ed esecuzione

Per poter eseguire la compilazione, si esegue il comando:

gcc server\_universitario.c -o server\_universitario

che genererà un file chiamato “server\_universitario”, eseguibile con il comando:

“./server\_universitario”



Da questo momento il server è in ascolto per le richieste sulla porta TCP 9002.

*CAPITOLO 2. MANUALE UTENTE*

## 2.2 - Segreteria

Si vanno a descrivere le istruzioni per compilare ed eseguire il codice relativo alla componente della segreteria.

### - Compilazione ed esecuzione

Per poter eseguire la compilazione, si esegue il comando:

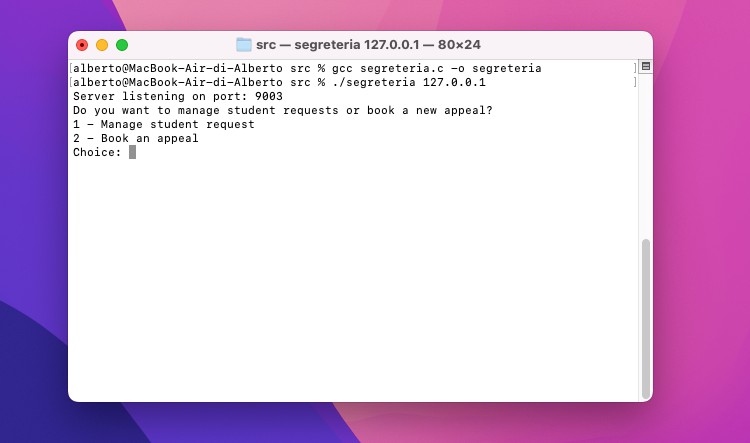
gcc segreteria.c -o segreteria

che genererà un file chiamato segreteria, eseguibile con il comando:

./segreteria <IP address>

Bisogna anche specificare l’indirizzo IP, che in fase di test sarà

127.0.0.1 (localhost).



La segreteria, come anticipato precedentemente, funge sia da client che da server. La scelta **1** permette di gestire tutte le richieste dei client “studente” (**server rispetto lo studente**);

La scelta **2** permette di inviare una richiesta di inserimento di un nuovo appello di un dato esame al server \_universitario (**client rispetto la segreteria**);

*CAPITOLO 2. MANUALE UTENTE*

## 2.3 - Studente

Si vanno a descrivere le istruzioni per compilare ed eseguire il codice relativo alla componente dello studente.

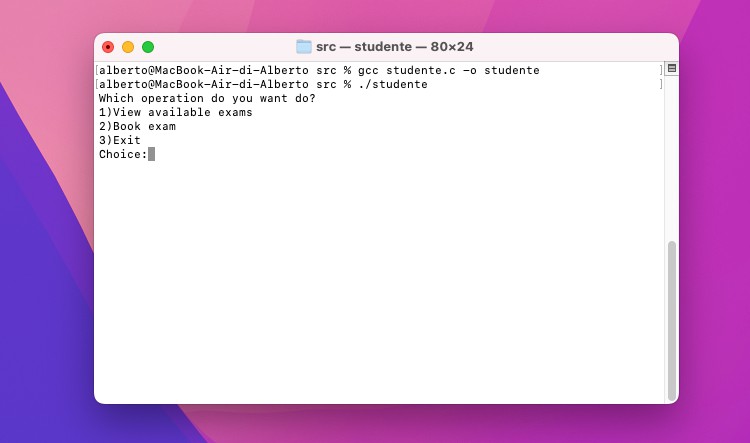
### - Compilazione ed esecuzione

Per poter eseguire la compilazione, si esegue il comando:

gcc studente.c -o studente

che genererà un file chiamato studente, eseguibile con il comando:

./studente



# Capitolo 3 – Casi D’uso

Di seguito sono mostrati i casi d’uso, con relativi esempi, dell’applicazione.

## 3.1 – La segreteria inserisce un nuovo appello per un esame

La segreteria sceglie di inserire un nuovo appello, fornisce il nome e la data, e viene inoltrata la richiesta al server che provvederà ad inserirlo nel file “exams.txt”

**Segreteria:**

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente



**Server universitario:**

Immagine che contiene testo, schermata, Sistema operativo, software

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

File exams.txt

Immagine che contiene schermata, testo, cornice

Descrizione generata automaticamente

*CAPITOLO 3. CASI D’USO*

## 3.2 – Lo studente chiede alla segreteria le date disponibili di un esame

Lo studente sceglie di richiedere le date disponibili di un esame, fornendo il nome dell’esame in input.

La segreteria inoltra la richiesta al server universitario che provvederà a restituire l’output.

**Studente**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



**Segreteria**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

**Server universitario**

Immagine che contiene testo, schermata, Sistema operativo, software

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

## 3.3 – Lo studente si prenota ad un appello

Lo studente sceglie di prenotarsi ad un appello, fornendo il nome dell’esame, la sua matricola e la data dell’appello in input.

La segreteria inoltra la richiesta al server universitario che provvederà a prenotarlo

(memorizzando l’informazione su un file), restituendo il numero progressivo della

prenotazione.

**Studente**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



**Segreteria**

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

**Server universitario**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



**File “bookings.txt”**

La prenotazione (NOME\_ESAME, MATRICOLA, DATA\_APPELLO, PROGRESSIVO) viene

memorizzata nel file “bookings.txt”

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

*CAPITOLO 3. CASI D’USO*

## 3.4 – Lo studente vuole prenotarsi ad un appello inesistente

Lo studente sceglie di prenotarsi ad un appello, fornendo il nome dell’esame, la sua matricola e la data dell’appello in input.

La segreteria inoltra la richiesta al server universitario che, non trovando l’appello nel file

“exams.txt”, restituirà un errore.

**Studente**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



**Segreteria**

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

**Server** **universitario**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

## 3.5 – La segreteria inserisce un appello già esistente

La segreteria sceglie di inserire un appello già esistente, fornendo il nome e la data.

Il server universitario, a seguito di opportuni controlli, restituisce un messaggio d’errore.

**Segreteria**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



**Server universitario**

Immagine che contiene testo, schermata, computer, Sistema operativo

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

## 3.6 – Lo studente tenta di prenotarsi due volte ad uno stesso appello

Lo studente sceglie di prenotarsi ad un appello due volte, fornendo il nome dell’esame, la sua matricola e la data dell’appello in input.

La segreteria inoltra la richiesta al server universitario che, a seguito di opportuni controlli, risponderà con un messaggio di errore.

**Studente**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



**Segreteria**

Immagine che contiene testo, schermata

Descrizione generata automaticamente



*CAPITOLO 3. CASI D’USO*

**Server universitario**

Immagine che contiene testo, schermata, computer, software

Descrizione generata automaticamente

