

UNIVERSITÀ DEGLI STUDI DI NAPOLI “PARTHENOPE”
FACOLTÀ DI SCIENZE E TECNOLOGIE
CORSO DI LAUREA IN INFORMATICA



PROGETTO DI RETI DI CALCOLATORI

“Università”

DOCENTE
Emanuel Di Nardo

STUDENTI
Alberto Gambalonga – 0124002583
Fabio Cosentino – 0124002490
Biagio De Blasio – XXXXXXXXXX

Anno Accademico 2023-2024

Indice

Traccia	3
Capitolo 1.....	4
1.1 Componenti coinvolte	5
Capitolo 2 - Manuale utente.....	7
2.1 Server universitario	8
2.1.1 Compilazione ed esecuzione	8
2.2 Segreteria.....	9
2.2.1 Compilazione ed esecuzione	9
2.3 Studente	10
2.3.1 Compilazione ed esecuzione	10

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.

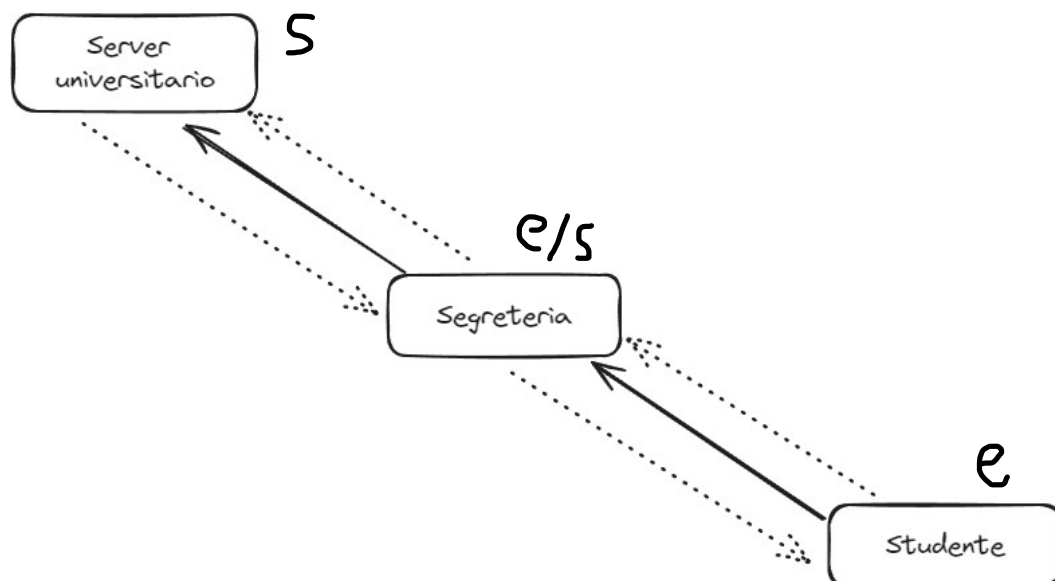


Figura 1.1: Diagramma architettura

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 richiestaviene effettuata dallo studente alla segreteria, la quale procede ad inoltrarla al server universitario. Inoltre, il server universitario invia alla segreteria il numero di prenotazione progressivo, il quale viene inoltrato allo studente;
 2. **Aggiungere un appello per un determinato esame:** tale richiesta viene effettuata dalla segreteria.
- **Segreteria:** questa componente funge da **server** per lo studente e da **client** per il server universitario, con compiti di:
 1. **Inviare al server universitario l'appello di un esame da aggiungere;**
 2. **Inoltrare la richiesta di prenotazione ad un appello,** effettuata da uno studente, al server universitario;
 3. **Fornire allo studente le date degli appelli di un esame specificato dallo studente.**
- **Studente:** questa componente funge da **client** per la segreteria, con compiti di:
 1. **Richiedere alla segreteria gli appelli di un esame da egli specificato;**
 2. **Inviare alla segreteria una richiesta di prenotazione ad undeterminato appello d'esame.**

CAPITOLO 1. DESCRIZIONE DEL PROGETTO E DELL'ARCHITETTURA⁴

Se durante le richieste effettuate la segreteria smette di rispondere, lo studente ritenta la connessione per tre volte, dopodichè, 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.

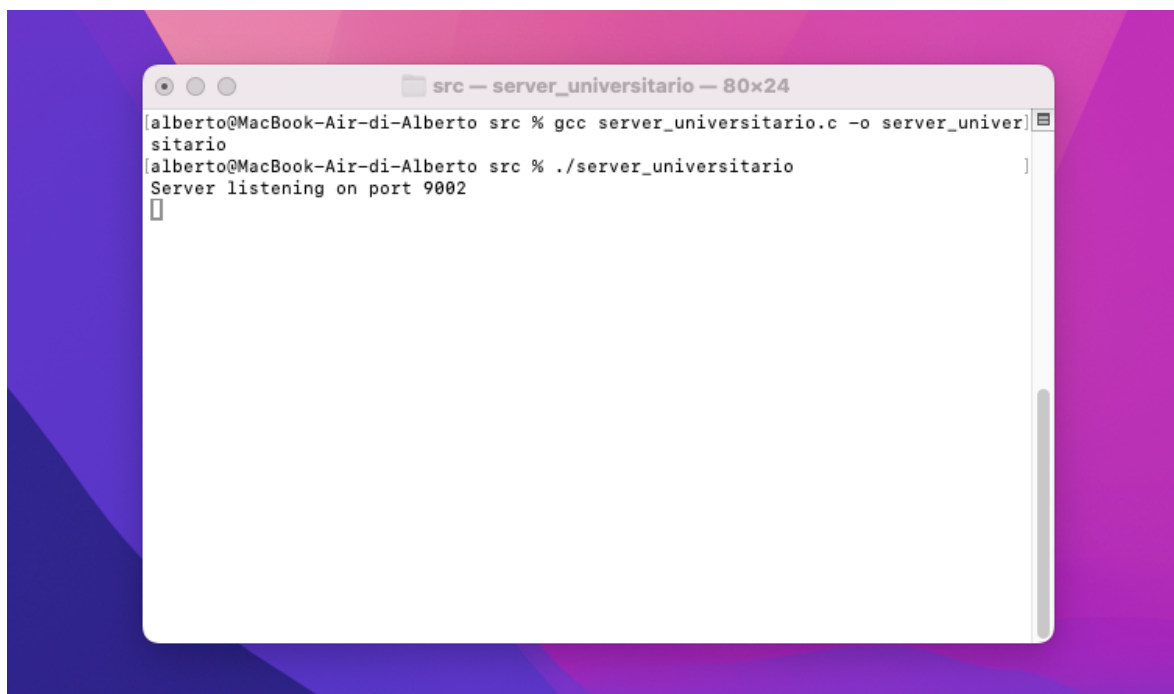
2.1.1 Compilazione ed esecuzione

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
```



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.

2.2.1 Compilazione ed esecuzione

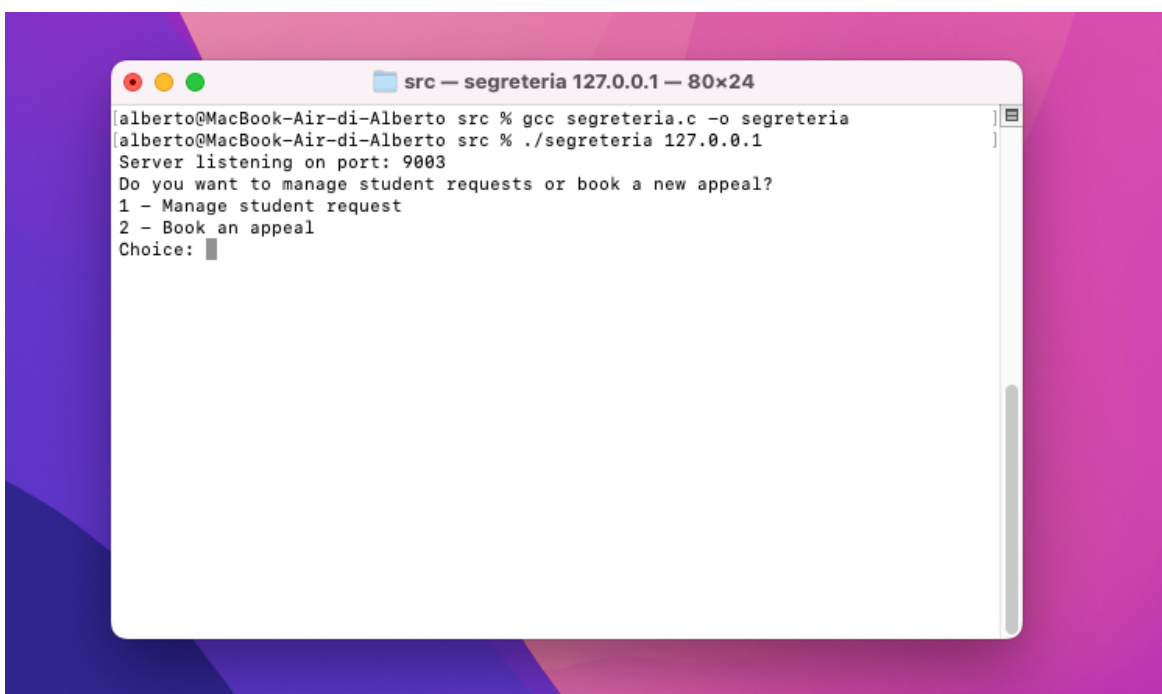
Per poter eseguire la compilazione, si esegue il comando:

```
gcc segreteria.c -o segreteria
```

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

```
./segreteria <IP address>
```

Inoltre, bisogna anche specificare l'indirizzo IP, che in fase di test sarà 127.0.0.1 (localhost).



```
src — segreteria 127.0.0.1 — 80x24
alberto@MacBook-Air-di-Alberto src % gcc segreteria.c -o segreteria
alberto@MacBook-Air-di-Alberto src % ./segreteria 127.0.0.1
Server listening on port: 9003
Do you want to manage student requests or book a new appeal?
1 - Manage student request
2 - Book an appeal
Choice: █
```

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**);

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

CAPITOLO 2. MANUALE UTENTE

2.3 Studente

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

2.3.1 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
```

