



UNIVERSITÀ
DEGLI STUDI
DI NAPOLI
PARTHENOPE

Progetto Reti di Calcolatori
Green Pass

Vincenzo Poli 0124001487
Vincenzo Lorenzo Poli, 0124000948

A.A. 2023/2024

Contents

1	Descrizione del Progetto	3
2	Descrizione dell'architettura	4
3	Dettagli implementativi dei client/server	5
4	Parti rilevanti del codice sviluppato	6
4.1	Centro Vaccinale	6
4.2	Server V	7
4.3	Server G	9
5	Manuale utente con le istruzioni su compilazione ed esecuzione	11

1 Descrizione del Progetto

Progettazione e gestione del servizio di green pass.

- 1) Avviato un client, che sarà quello con cui l'utente si interfacerà, verrà richiesto il codice della propria tessera sanitaria, informazione poi inoltrata ad un centro vaccinale.
- 2) Il centro vaccinale terrà conto della data di inserimento per stabilirne il periodo di validità, e comunicherà queste info ad un primo server.
- 3) Un secondo client avrà la funzione di verificare la validità dello stesso green pass.
- 4) Un terzo client invece terrà conto dei contagi.
- 5) Un secondo server si interfacerà a questi altri due client per offrire il servizio di verifica di validità dei green pass. Il secondo server interagirà col primo per implementare questo servizio.

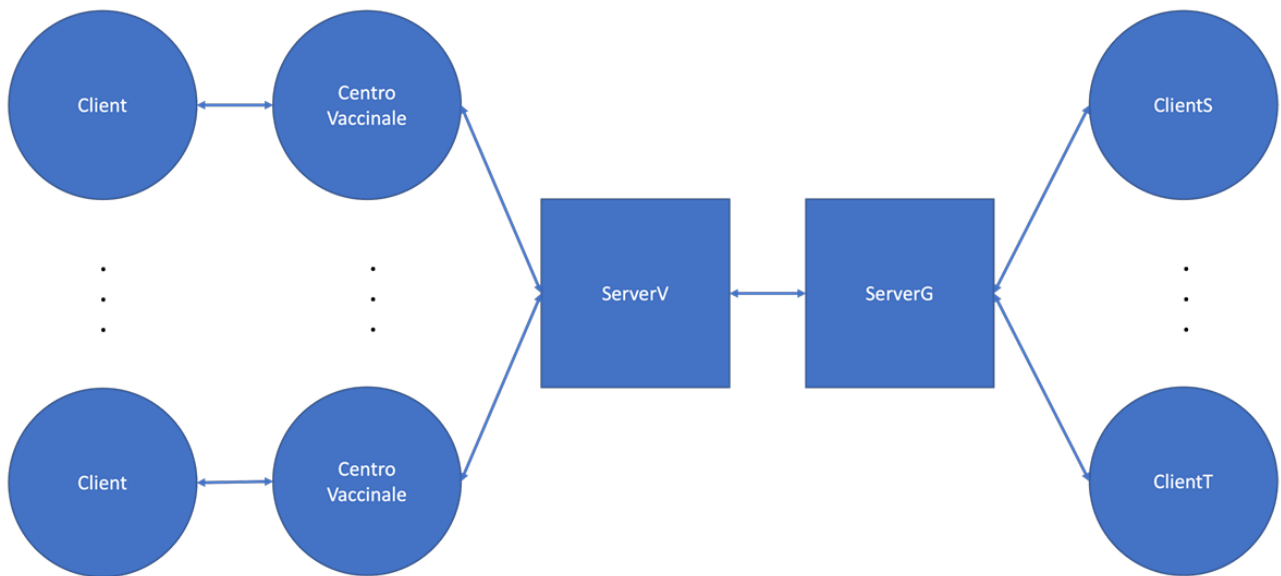
2 Descrizione dell'architettura

Il primo client si interfaccia al centro vaccinale, come da descrizione, ed è nominato client. Il primo server - quello a cui verranno inoltrati i nuovi vaccinati - è chiamato server V, il secondo server - quello che offre il servizio di verifica della validità dei green pass - è chiamato server G.

Il client che verifica la validità dei green pass è chiamato client S, quello che si occupa dei contagi è chiamato Client T.

Dallo schema è possibile notare che:

- 1) Nessun client interagisce direttamente con il server V.
- 2) Ne consegue una struttura a "più strati" che obbliga una gestione non banale di connessioni multiple sia in fase di attesa delle richieste, sia per fornire le relative risposte.



3 Dettagli implementativi dei client/server

Utilizzo del linguaggio C e delle sue librerie per le socket.

- 1) Il centro vaccinale si mette in ascolto lato client e tenta di connettersi al server V dal lato opposto. Uso della concorrenza per l'invio dei dati ricevuti dal client al server V.
- 2) Il server V deve comunicare sia col centro vaccinale che con il server G, e per ognuno di essi implementare il servizio richiesto. Per la gestione dei canali di comunicazione si è ricorsi al multiplexing. Per la gestione dei descrittori (e quindi l'implementazione dei servizi richiesti) si è invece utilizzata la concorrenza.
- 3) Il server G gestisce le interazioni col client S ed il client T in maniera non dissimile dal server V rispetto al centro vaccinale ed al server G. Si usa quindi multiplexing e concorrenza, in particolare la comunicazione tra server V e server G è delegata ad un figlio (creato allo sbloccarsi della funzione select) che si occuperà della gestione del descrittore del client S.
- 4) I client sono molto meno complessi rispetto ai due server, e si avvalgono delle socket per semplici tentativi di connessione. Vale però la pena approfondire il funzionamento del client S: richiede l'interazione con l'utente e l'attesa di una risposta dal server G. Per lo scopo, si è di nuovo utilizzato il multiplexing.

4 Parti rilevanti del codice sviluppato

4.1 Centro Vaccinale

```
for(;;)
{
    /*accetta una connessione su listen fd da parte di un client
    ed apre un nuovo canale di comunicazione con quel client su connfd*/
    connfd = Accept(listenfd, (struct sockaddr *) NULL, NULL);

    //legge i dati inviati dal client
    int bytesRead = FullRead(connfd, &buf, BUF_SIZE);
    //int bytesRead = read(connfd, &buf, BUF_SIZE);

    if (bytesRead == 0)
    {
        //Il client ha chiuso la connessione
        close(connfd);
    }
}
```

Figure 1: Centro Vaccinale, controllo sui byte letti

```
else if (bytesRead > 0)
{
    if ((pid = fork()) == 0)
    {
        //sono il processo figlio

        //il figlio non deve rimanere in ascolto, deve invece comunicare i dati inviati dal client al serverV
        close(listenfd);

        //imposta l'ultimo carattere di fine stringa (o EOF?)
        buf[bytesRead]=0;
        strcpy(tmpGreenPass.code, buf);
        tmpGreenPass.valid = VALIDITA_GREENPASS;
        tmpGreenPass.t = time(NULL);

        //dopo aver inserito le informazioni (passo precedente) ricevute dalla struttura GREENPASS temporary, stampa quelle info sul terminale
        printf("Ricevuto nuovo Green pass %s, valido per %d giorni...lo invio a %d\n", tmpGreenPass.code, tmpGreenPass.valid, servVfd);
        //invia questa struttura al server
        FullWrite(servVfd, &tmpGreenPass, sizeof(GreenPass));

        printf("ho fatto la full write.. ho scritto %ld byte \n", sizeof(GreenPass));

        //chiude canale di comunicazione col client
        close(connfd);
        exit(0);
    }
}
```

Figure 2: Centro Vaccinale, concorrenza

4.2 Server V

```
for(;;)
{
    //inizializza il readset a tutti 0
    FD_ZERO(&readset);
    //imposta il readset aggiungendo il descrittore per il CV
    FD_SET(listenfd_cv, &readset);
    //imposta il readset aggiungendo il descrittore per il serverG
    FD_SET(listenfd_sg, &readset);
    //funzione max per ricevere il FD più grande
    maxfd = GetMaxFDPl();
}
```

Figure 3: Server V, inizializzazione readset

```
if (Select(maxfd, &readset, NULL, NULL, NULL) > 0)
{
    //se ci si è sbloccati per una richiesta dal CV
    if (FD_ISSET(listenfd_cv, &readset))
    {
        //si accetta la connessione e si apre canale di comunicazione col CV
        connfd_cv = Accept(listenfd_cv, (struct sockaddr *) NULL, NULL);

        //il figlio gestirà il servizio per il CV
        if ((pid = fork()) == 0)
        {
            /*sono il processo figlio

            il figlio non deve rimanere in ascolto, deve invece scrivere su file i dati ricevuti dal CV*/
            close(listenfd_cv);
            //funzione che gestisce il servizio precedentemente citato
            GestioneDescrittoreCentroVaccinale();
            //chiude canale di comunicazione col CV
            close(connfd_cv);
            exit(0);
        }
    }
}
```

Figure 4: Server V, gestione centro vaccinale

```

//se ci si è sbloccati per una richiesta dal SG
if (FD_ISSET(listenfd_sg, &readset))
{
    //si accetta la connessione e si apre canale di comunicazione col serverG
    connfd_sg = Accept(listenfd_sg, (struct sockaddr *) NULL, NULL);

    if ((pid = fork()) == 0)
    {
        /*sono il processo figlio

        il figlio non deve rimanere in ascolto, deve invece occuparsi del servizio relativo al serverG*/
        close(listenfd_sg);
        //funzione che gestisce il servizio precedentemente citato
        GestioneDescrittoreServerG();
        //chiude canale di comunicazione col SG
        close(connfd_sg);
        exit(0);
    }
}

```

Figure 5: Server V, gestione server G

4.3 Server G

```
for(;;)
{
    //inizializza il readset a tutti 0
    FD_ZERO(&readset);
    //imposta il readset aggiungendo il descrittore per il ClientT
    FD_SET(listenfd_ct, &readset);
    //imposta il readset aggiungendo il descrittore per il ClientS
    FD_SET(listenfd_cs, &readset);
    //funzione max per ricevere il FD più grande
    maxfd = GetMaxFDPl();
}
```

Figure 6: Server G, inizializzazione readset

```
if (Select(maxfd, &readset, NULL, NULL, NULL) > 0)
{
    //se CS si è sbloccati per una richiesta dal ClientT
    if (FD_ISSET(listenfd_ct, &readset))
    {
        //il clientT è in grado di validare / invalidare un green pass a seconda se CS comunica che l'utente è contagiato o meno
        //si accetta la connessione e si apre canale di comunicazione col CT
        connfd_ct = Accept(listenfd_ct, (struct sockaddr *) NULL, NULL);

        //il figlio gestirà il servizio per il CT
        if ((pid = fork()) == 0)
        {
            /*sono il processo figlio

            il figlio non deve rimanere in ascolto, deve invece scrivere su file i dati ricevuti dal CT*/
            close(listenfd_ct);
            close(listenfd_cs);
            //funzione che gestisce il servizio precedentemente CStato
            GestioneDescrittoreClientT();
            //chiude canale di comunicazione col CT
            close(connfd_ct);
            exit(0);
        }
    }
}
```

Figure 7: Server G, gestione client T

```

//se CS si è sbloccati per una richiesta dal ClientS
if (FD_ISSET(listenfd_cs, &readset))
{
    //il ClientT è in grado di validare / invalidare un green pass a seconda se CS comunica che l'utente è contagiato o meno
    //si accetta la connessione e si apre canale di comunicazione col CS
    connfd_cs = Accept(listenfd_cs, (struct sockaddr *) NULL, NULL);

    //il figlio gestirà il servizio per il CS
    if ((pid = fork()) == 0)
    {
        /*sono il processo figlio

        il figlio non deve rimanere in ascolto, deve invece scrivere su file i dati ricevuti dal CS*/
        close(listenfd_ct);
        close(listenfd_cs);
        //funzione che gestisce il servizio precedentemente citato
        GestioneDescrittoreClientS();
        //chiude canale di comunicazione col CS
        close(connfd_cs);
        exit(0);
    }
}

```

Figure 8: Server G, gestione client S

Nella funzione di gestione del client S è presente una ulteriore chiamata a funzione. Quella funzione gestirà la comunicazione col server V necessaria per la verifica di validità del green pass.

5 Manuale utente con le istruzioni su compilazione ed esecuzione

- 1) Il progetto include una cartella contenente file di script numerati.
- 2) Forniti i permessi di lettura e scrittura necessari per ogni file di script, è sufficiente eseguirli nel terminale seguendo la numerazione.
- 3) È preferibile inserire i codici fiscali dei vaccinati e/o contagiati prima di verificare la validità del green pass associato: questa accortezza permette la creazione dei file da cui i processi leggeranno per fornire le risposte necessarie.
- 4) I processi aperti saranno sei.
- 5) In testa saranno stampati a video i nomi dei processi avviati.
- 6) Nella pagina successiva sarà fornita un esempio di esecuzione.

```
Terminale
File Modifica Visualizza Cerca Terminale Aiuto
...
Ricevuti 40 byte...
Ricevuto nuovo Green pass DDD111, valido per 180 giorni
...
E' arrivata una richiesta dal SG
Ricevuti 7 byte...
Lettura numero 0
Letto green pass.. Codice tessera: AAA111
return 1
Lettura numero 1
Letto green pass.. Codice tessera: BBB111
return 0
Green pass valido: 1
█

Terminale
File Modifica Visualizza Cerca Terminale Aiuto
Avvio il centro vaccinale...
*** CENTRO_VACCINALE *****
- Centro vaccinale in ascolto su porta 5033
- Ricevuto nuovo Green pass AAA111, valido per 180 giorni.
  lo invio a ServerV
- Ricevuto nuovo Green pass BBB111, valido per 180 giorni.
  lo invio a ServerV
- Ricevuto nuovo Green pass CCC111, valido per 180 giorni.
  lo invio a ServerV
- Ricevuto nuovo Green pass DDD111, valido per 180 giorni.
  lo invio a ServerV

Terminale
File Modifica Visualizza Cerca Terminale Aiuto
1...
Il client T ha chiuso la connessione...
Ricevuti 7 byte... codice tessera AAA111...
Il client S ha chiuso la connessione...
Ricevuti 7 byte... codice tessera BBB111...
No dati da leggere
Connesso al server V
Il codice della tessera: BBB111, è stata inviata al serverV. Numero caratteri inviati: 7
Il server V ha detto che BBB111 ha un green pass VALID
0
Il client S ha chiuso la connessione...
█

Terminale
File Modifica Visualizza Cerca Terminale Aiuto
Avvio il client...
***** CLIENT *****
Inserire il codice di tessera sanitaria:
AAA111
Inserire il codice di tessera sanitaria:
BBB111
Inserire il codice di tessera sanitaria:
CCC111
Inserire il codice di tessera sanitaria:
DDD111
Inserire il codice di tessera sanitaria:
█

Terminale
File Modifica Visualizza Cerca Terminale Aiuto
Avvio il clientT...
***** CLIENTT *****
Inserire il tuo codice di tessera sanitaria:
AAA111
Inserire [1] se l'utente è contagiato, [0] se l'utente è guarito:
1
Inserito codice AAA111: Contagiato: SI
Inserire il tuo codice di tessera sanitaria:
█

Terminale
File Modifica Visualizza Cerca Terminale Aiuto
Avvio il clientS...
***** CLIENTS *****
1 - Inserire il codice di tessera sanitaria:
AAA111
da buffer di tastiera ho letto 6 caratteri
2 - GREEN PASS NON VALIDO
1 - Inserire il codice di tessera sanitaria:
BBB111
da buffer di tastiera ho letto 6 caratteri
2 - GREEN PASS VALIDO
1 - Inserire il codice di tessera sanitaria:
```

Figure 9: Un esempio di esecuzione: inserimento codice fiscale di 4 utenti vaccinati e contestuale verifica di validità del green pass