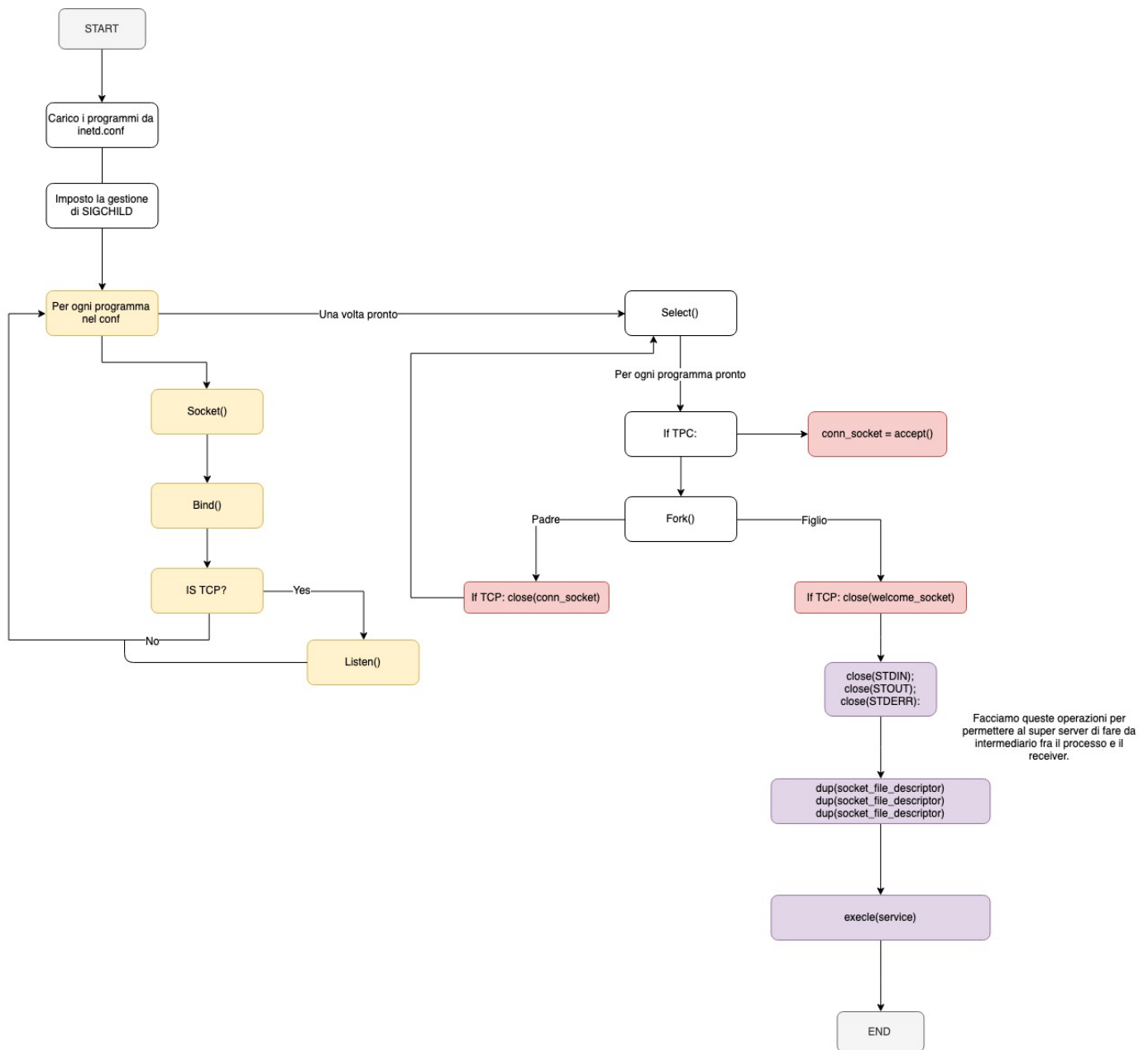


# Assignment 2

## Task 1



Per vederlo ingrandito, consigliamo di consultare [questo link](#).

## Task 2

Il superserver e' stato costruito partendo dallo schema in maniera modulare.  
La struttura prevede diversi passaggi:

- lettura della lista dei servizi dal file `inetd.conf`
- inizializzazione delle socket per i diversi servizi
- gestione della select e della creazione dei processi per ogni chiamata a servizio

Il sistema si compone di diversi moduli che sono stati integrati per migliorare la comprensione del codice:

- l'header file `service.h` include la dichiarazione della struct `service`
- l'header file `init_services.h` include la dichiarazione della funzione che estrae la lista di servizi da un file
- il file `init_services.c` contiene l'implementazione della funzione sopra citata
- l'header file `checkers.h` contiene funzioni di utility per verificare alcuni parametri, come ad esempio se il `transport_protocol` è `tcp` o `udp`, ecc.

Per ogni funzione è presente sufficiente documentazione su parametri in input, ciò che computa e valori di ritorno.

Per la compilazione del codice è stato scritto un makefile che, una volta eseguito con il comando `make` , genera il file eseguibile `superserver.exe` . Per lanciarlo è sufficiente lanciare, all'interno della directory in cui è collocato, il comando `./superserver.exe` .

Nota importante: per poter eseguire correttamente il makefile è necessario aver installato il compilatore **gcc**, utilizzato per le fasi di compilazione e linking.

I test che sono stati eseguiti sono i seguenti:

- test sulla correttezza della funzione `init_services` presente nel file `init_services.c` testando la lettura di diversi input file con errori e non
- test sulla correttezza delle funzioni presenti in `checkers.h`
- test sulla corretta esecuzione di `superserver` :
  - test utilizzando `udpServer` e `tcpServer` in modalità `wait` e `nowait` con connessioni multiple da parte di più client in `localhost`
  - test utilizzando `udpServer` e `tcpServer` con client che aprono connessioni a un server su una macchina diversa sulla stessa rete

Per quanto riguarda la differenza tra modalità `wait` e `nowait` sono state registrate le seguenti informazioni:

-	udpServer	tcpServer
wait	Alla prima richiesta da parte di un client viene creato un nuovo processo che risponde al client. Da lì in poi per ogni nuova connessione da parte di altri client viene sempre utilizzato lo stesso processo.	Alla prima richiesta di connessione, essa viene accettata dal superserver e viene creato un nuovo processo. Le successive richieste aspetteranno che la connessione aperta venga chiusa prima di poter essere accettate dal superserver.
nowait	Per ogni nuova richiesta viene creato un nuovo processo che risponde al client che l'ha inviata	Per ogni nuova richiesta verso il superserver, esso crea una nuova socket riservata alla richiesta e con essa viene istanziato un nuovo processo che risponde ad un solo client. Perciò l'attesa che si verificava in modalità wait non è più presente.

## Terminal Screenshots

### Note importanti

- per questo test si sono modificati `udpServer` e `tcpServer` in modo da ritornare al client il pid del processo.
- Le porte che sono stampate dal server corrispondono a:
  - 8501: `udpServer` in wait mode
  - 8502: `tcpServer` in wait mode
  - 8503: `udpServer` in nowait mode
  - 8504: `tcpServer` in nowait mode
- nell'utilizzo di socket TCP viene inviato il messaggio "Undefined error: 0". Ciò sembra relativo a macOS in quanto su macchine Linux non si verifica. Questo messaggio non inficia il corretto invio del contenuto al client (negli Screenshot il pid).

## udpServer in wait mode

```

ggobb1q38f9d36a2c0f ~/Desktop/Assignment2/assignment2/assignment2 master ➤ ./superserver.exe
e "{\"HOME=/usr/home\", \"LOGNAME=home\", (char *)0}"
Opened udp socket on IP: 127.0.0.1 and port: 8501 in wait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8502 in wait mode
Opened udp socket on IP: 127.0.0.1 and port: 8503 in nowait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8504 in nowait mode
Created process 28021

ggobb1q38f9d36a2c0f ~/Desktop/Assignment2/assignment2/assignment2 master ➤ ./udpClient.exe 127.0.0.1 8501
Insert message:
hello
String going to be sent to server: hello
Bytes sent to server: 5
Received from server: 28021
Insert message:

```

Come è possibile notare, sono state inviate due richieste da due client diversi ed entrambi sono stati serviti dalla stessa istanza di `udpServer` (pid 8028).

## tcpServer in wait mode

```

ggobb1q38f9d36a2c0f ~/Desktop/Assignment2/assignment2/assignment2 master ➤ ./superserver.exe
e "{\"HOME=/usr/home\", \"LOGNAME=home\", (char *)0}"
Opened udp socket on IP: 127.0.0.1 and port: 8501 in wait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8502 in wait mode
Opened udp socket on IP: 127.0.0.1 and port: 8503 in nowait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8504 in nowait mode
Created process 28586

ggobb1q38f9d36a2c0f ~/Desktop/Assignment2/assignment2/assignment2 master ➤ ./tcpClient.exe 127.0.0.1 8502
Insert message:
hello
String going to be sent to server: hello
Bytes sent to server: 5
Received from server: : Undefined error: 0
28586
Insert message:

ggobb1q38f9d36a2c0f ~/Desktop/Assignment2/assignment2/assignment2 master ➤ ./tcpClient.exe 127.0.0.1 8502
Insert message:
hello
String going to be sent to server: hello
Bytes sent to server: 5

```

Come già descritto precedentemente, una volta accettata una connessione e istanziato un processo, le altre richieste rimangono in attesa di essere servite. Verranno servite solamente quando la connessione aperta verrà chiusa.

## udpServer in nowait mode

```
ggobbio@38f9d36a2c0f: ~/Desktop/Assignment2/assignment2 ➤ master ➤ ./superserver.exe
e {"HOME=/usr/home", "LOGNAME=home", (char *)0}"
Opened udp socket on IP: 127.0.0.1 and port: 8501 in wait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8502 in wait mode
Opened udp socket on IP: 127.0.0.1 and port: 8503 in nowait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8504 in nowait mode
Created process 31806
Created process 31810
[]

ggobbio@38f9d36a2c0f: ~/Desktop/Assignment2/assignment2 ➤ master ➤ ./udpClient.exe 127.0.0.1 8503
Insert message:
hello
String going to be sent to server: hello
Bytes sent to server: 5
Received from server: 31806
Insert message:
[]

ggobbio@38f9d36a2c0f: ~/Desktop/Assignment2/assignment2 ➤ master ➤ ./udpClient.exe 127.0.0.1 8503
Insert message:
hello
String going to be sent to server: hello
Bytes sent to server: 5
Received from server: 31806
Insert message:
[]
```

Come è possibile notare, sono state inviate due richieste da due client diversi, perciò sono state create due istanze di udpServer. L'istanza che risponde è comunque una casuale tra quelle in attesa.

## tcpServer in nowait mode

```
ggobbio@38f9d36a2c0f: ~/Desktop/Assignment2/assignment2 ➤ master ➤ ./superserver.exe
e {"HOME=/usr/home", "LOGNAME=home", (char *)0}"
Opened udp socket on IP: 127.0.0.1 and port: 8501 in wait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8502 in wait mode
Opened udp socket on IP: 127.0.0.1 and port: 8503 in nowait mode
Opened tcp socket on IP: 127.0.0.1 and port: 8504 in nowait mode
Created process 30654
Created process 30858
[]

ggobbio@38f9d36a2c0f: ~/Desktop/Assignment2/assignment2 ➤ master ➤ ./tcpClient.exe 127.0.0.1 8504
Insert message:
hello
String going to be sent to server: hello
Bytes sent to server: 5
Received from server: : Undefined error: 0
30654
Insert message:
[]

ggobbio@38f9d36a2c0f: ~/Desktop/Assignment2/assignment2 ➤ master ➤ ./tcpClient.exe 127.0.0.1 8504
Insert message:
hello
String going to be sent to server: hello
Bytes sent to server: 5
Received from server: : Undefined error: 0
30858
Insert message:
[]
```

Sono state effettuate due richieste al server da parte di due client diversi, e per ognuno di essi è stata creata una nuova istanza di tcpServer che è riservata esclusivamente alla comunicazione con un solo client.