

ESERCITAZIONE 4

Server multiservizio: Socket C con select

Bernardi Daniel

Chichifoi Karina

Gjura Endri

Ivan Andrei Daniel

Pizzini Cavagna Hiari

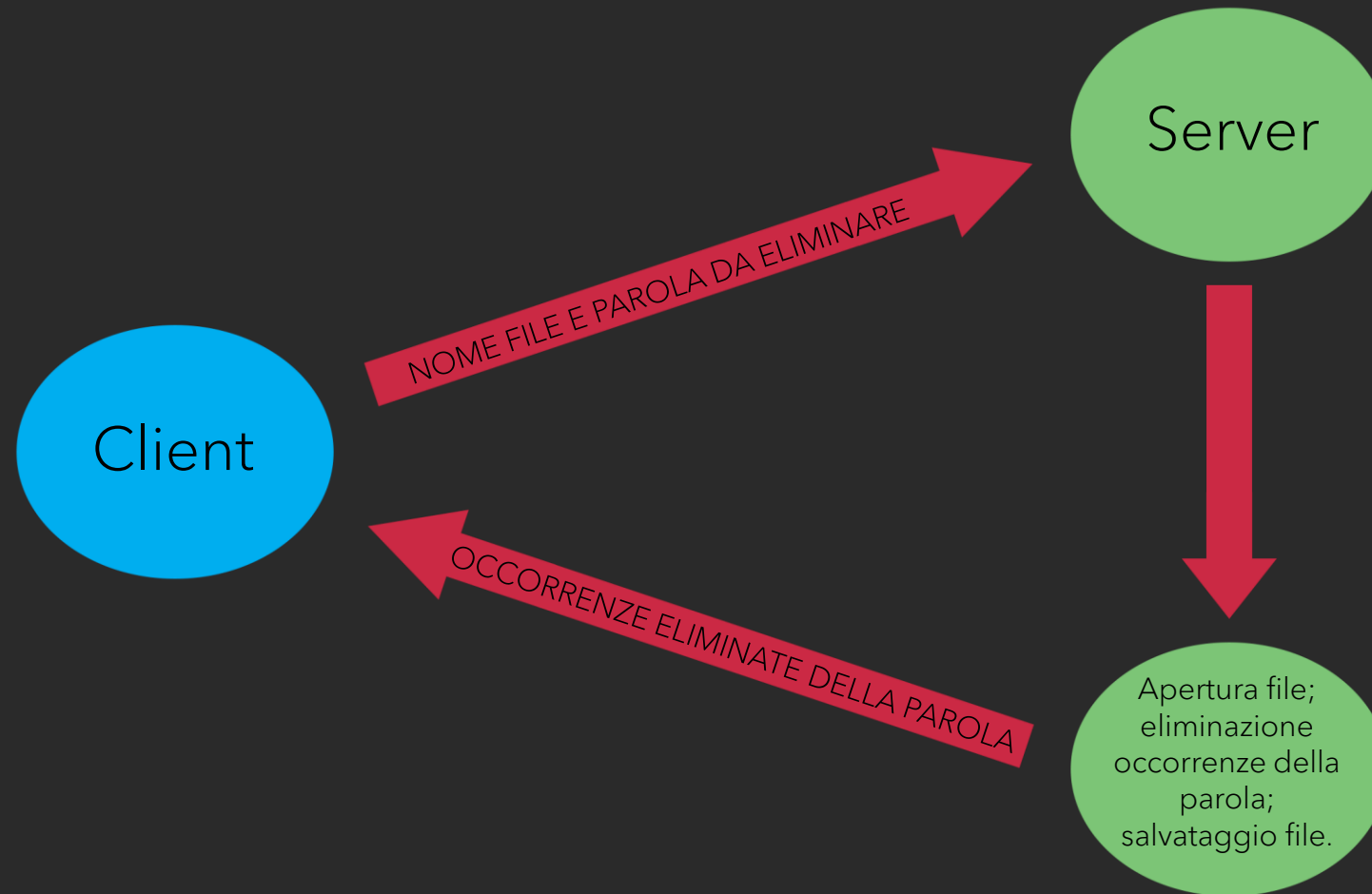
Introduzione

Per il C/S **UDP** si è considerato unicamente il caso in cui la parola sia preceduta e finisca con uno spazio o una nuova linea.

Viene restituito un intero negativo nel caso di file non trovato oppure il numero di occorrenze eliminate.

Per il C/S **TCP** invece il Server invia un esito positivo al Client se la directory passata viene aperta correttamente e successivamente il Client si mette in ascolto dei nomi dei file, altrimenti richiede una nuova directory.

Schema C/S senza connessione (UDP)

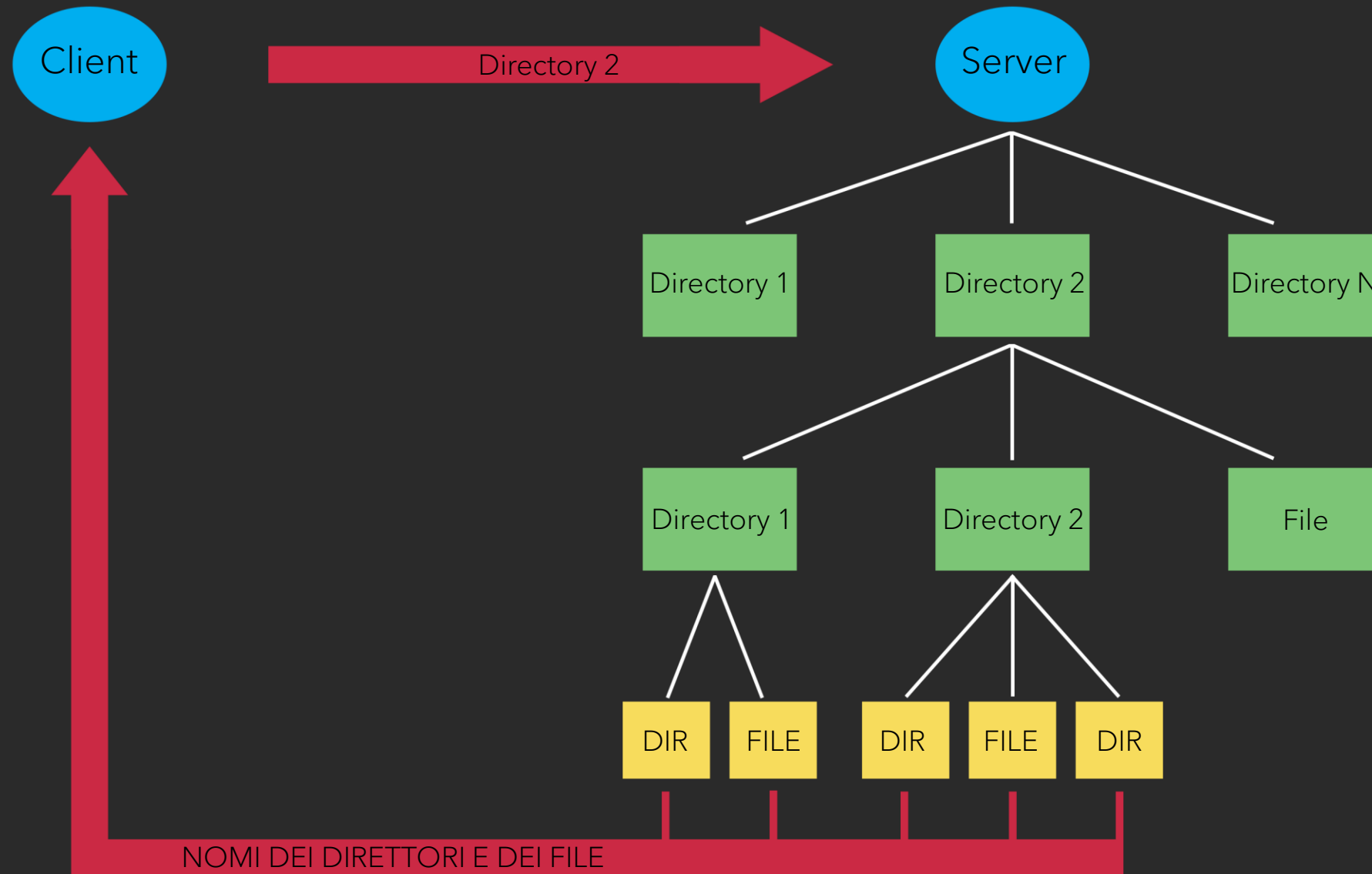


Algoritmo Server (UDP)

```
len_word = strlen(request.word);
while ((nread = read(fd_fileUDP_in, &buff, DIM_BUFF)) > 0) {
    check_word = 0;
    for (i = 0; i < nread; i++) {
        j = 0;
        if (check_word == 0
            && ((i + len_word) < nread)
            && (buff[i+len_word] == ' ' || buff[i+len_word] == '\n')) {
            while (j < len_word && buff[i + j] == request.word[j]) {
                j++;
            }
        }
        if (j != len_word) {
            write(fd_fileUDP_out, &(buff[i]), sizeof(char));
        }
        else {
            i += len_word;
            ris++;
        }
        if (buff[i] == ' ' || buff[i] == '\n') {
            check_word = 0;
        }
        else {
            check_word = 1;
        }
    }
}
```

Algoritmo di rimozione parola. Andrà a riscrivere su un file d'appoggio, eliminando le occorrenze solo nel caso siano circondate da spazi o da nuove linee

Schema C/S con connessione (TCP)



Algoritmo Server (TCP) 1

```
While ((ent = readdir(dir)) != NULL) {  
    if(ent->d_name[0] == '.') {  
        if (ent->d_name[1] == '.') {  
            if (ent->d_name[2] == '\\0') {  
                continue;  
            }  
        }  
        if (ent->d_name[1] == '\\0') {  
            continue;  
        }  
    }  
  
    strcat(dirBuff, "/");  
    strcat(dirBuff, ent->d_name);  
}
```

Il Server legge ciclicamente il contenuto della directory ricevuta dal Client e prepara un buffer contenente il path dei file interni alla directory.

Algoritmo Server (TCP) 2

```
stat(ent->d_name, &path_stat);
if (S_ISDIR(path_stat.st_mode) == 0) {
    if ((tdir = opendir(dirBuff)) != NULL) {
        while ((ent = readdir(tdir)) != NULL) {
            if (ent->d_name[0] == '.') {
                if (ent->d_name[1] == '.') {
                    if (ent->d_name[2] == '\0') {
                        continue;
                    }
                }

                if (ent->d_name[1] == '\0') {
                    continue;
                }
            }
            write(connfd, ent->d_name, MAX_LENGTH);
        }
        closedir(tdir);
    }
    strcpy(dirBuff, dirName);
}
```

Viene controllato che il file nel buffer sia una directory; se lo è, viene esplorata e viene inviato al Client il nome dei file contenuti al suo interno.