

12 Ottobre 2021



Bernardini Claudio
Corsetti Luca
Giardina Gianluca
Straccali Leonardo

Esercitazione 0

File in JAVA e C

Obiettivi esercitazione

Hard skills

1

Lettura di un file in JAVA e C

2

Realizzazione di un filtro a carattere

Soft skills

1

Collaborazione per raggiungere un fine comune

2

Organizzazione del proprio lavoro in team

Introduzione

Produttore

1

Lettura delle linee passate dall'utente

2

Apertura del file condiviso

3

Scrittura su file delle
linee passate dall'utente

Consumatore

1

Lettura a carattere del file

2

Rimozione in output caratteri
passati come parametro

3

Stampa contenuto post filtraggio

3

Implementazione

C - Produttore

```
/**
 * Lettura righe dall'utente
 * Scrittura delle righe su file appena aperto
 */
printf("Inserisci le righe del file: [EOF per terminare l'inserimento] \n");
while (gets(riga) != NULL) {
    /* la gets legge tutta la riga, separatori inclusi, e trasforma il fine
    linea in fine stringa */
    riga[strlen(riga)+1]='\0';
    riga[strlen(riga)]='\n'; // aggiungo il fine linea
    written = write(fd, riga, strlen(riga)); // uso della primitiva
    if (written < 0){
        perror("P0: errore nella scrittura sul file");
        exit(3);
    }
}
```

JAVA - Produttore

```
in = new BufferedReader(new InputStreamReader(System.in));

FileWriter fout = null;
String inputl = null;
try {
    fout = new FileWriter(args[0]);
    System.out.println("Inserisci linee del file - [EOF per terminare]:");
    while ((inputl = in.readLine()) != null) {
        inputl += System.lineSeparator();
        fout.write(inputl, 0, inputl.length());
    }
    fout.close();
} catch (NumberFormatException nfe) {
    nfe.printStackTrace();
    System.exit(1); // uscita con errore, intero positivo a livello di sistema Unix
} catch (IOException e) {
    e.printStackTrace();
    System.exit(2); // uscita con errore, intero positivo a livello di sistema Unix
}
```

Implementazione

C - Consumatore

```
count = 0;
dimS = strlen(argv[1]); //numero caratteri da rimuovere
for (int i = 0; i < dimS; i++)
{
    tempC = toupper(argv[1][i]);
    if (strchr(delete_chars, tempC) == NULL)
    {
        delete_chars[count] = tempC;
        count++;
    }
}
delete_chars[count] = '\0';

while ((nread = read(fd, &read_char, sizeof(char))) /* Fino ad EOF*/
{
    if (nread >= 0)
    {
        if ((strchr(delete_chars, toupper(read_char))) == NULL) //Verifico che carattere
            putchar(read_char);
        }
    else
    {
        printf("(PID %d) impossibile leggere dal file %s", getpid(), file_in);
        perror("Errore!");
        close(fd);
        exit(3);
    }
}
close(fd);
```

JAVA - Consumatore

```
try {
    // se args.length == 2 allora il filtro è passato come parametro insieme al file
    // altrimenti il file è passato mediante ridirezione input
    if (args.length == 2) {
        r = new FileReader(args[1]);
    } else {
        r = new FileReader(FileDescriptor.in);
    }
} catch (FileNotFoundException e) {
    System.out.println("File non trovato");
    System.exit(1);
}

try {
    while ((x = r.read()) >= 0) {
        ch = (char) x;
        // se il carattere non è tra quelli da filtrare, printa
        if (filter.indexOf(ch) == -1)
            System.out.print(ch);
    }
    r.close();
} catch (IOException ex) {
    System.out.println("Errore di input");
    System.exit(2);
}
```

Gestione progetto

Coding



git



Team

