Esercizio

Realizzare un programma C che, utilizzando le system call di UNIX, preveda un'interfaccia del tipo:

```
esame   F   N   C
```

dove:

- F rappresenta il nome assoluto di un file
- N rappresenta un intero
- C rappresenta un carattere.

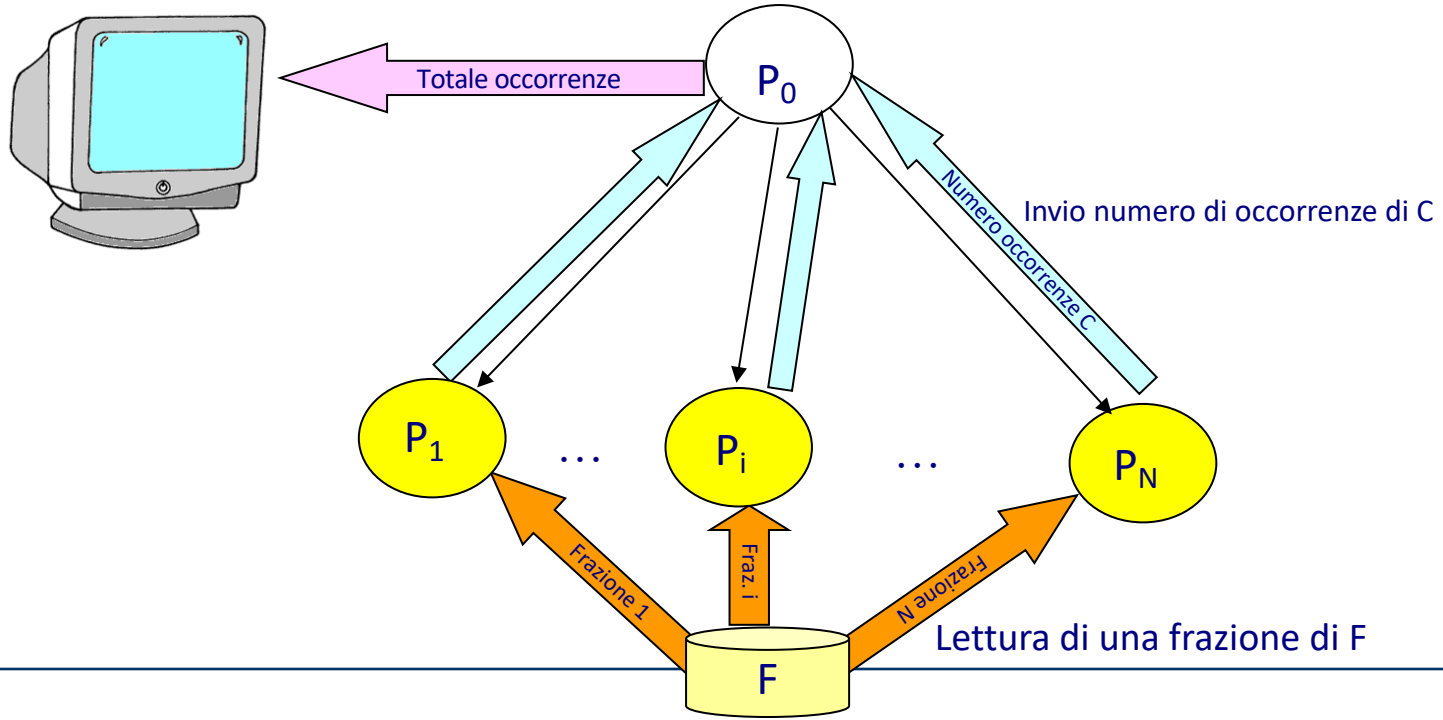
Il processo iniziale P_0 deve creare un numero N di processi figli (P_1, P_2, .. P_N).

Ogni figlio P_i (i=1,.. N) deve leggere una parte diversa del file F: in particolare, se L è la lunghezza del file F, ogni figlio dovrà leggere una frazione di L/N caratteri dal file F, secondo il seguente criterio:

- P1 leggerà la prima frazione;
- P2 leggerà la seconda frazione;
- ...
- PN leggerà l'ultima frazione N;

Ogni processo Pi leggerà quindi una frazione di F allo scopo di calcolare il numero delle occorrenze del carattere C nella parte di file esaminata; al termine della scansione, Pi comunicherà al padre il numero delle occorrenze di C incontrate nella frazione di file assegnatagli.

Il padre P0, una volta ottenuti i risultati da tutti i figli, stamperà il numero totale di occorrenze di C nel file F e terminerà.



Lettura di una frazione di F

Impostazione

- Accesso concorrente di processi a parti diverse dello stesso file: aperture separate del file da parte dei figli per evitare la condivisione dell'I/O pointer.

- Comunicazione dei figli con il padre: uso di pipe. Quante? Il padre non ha bisogno di distinguere i mittenti dei messaggi → è sufficiente una sola pipe.

- Messaggi: ogni messaggio rappresenta un intero → scrittura/lettura binaria verso/da pipe (v. file «binari»)

- Calcolo della dimensione di un file e posizionamento dell'I/O pointer in posizione arbitraria: uso di lseek()