

Esercizio 1

(15 min.)

Scrivere un programma C in ambiente Windows che prende inizialmente una stringa da input (può contenere anche spazi bianchi) e la salva in un buffer, per poi creare un processo figlio che manda in stampa la stessa stringa acquisita dal processo padre.

In più è richiesto che il processo padre termini solo dopo che il processo figlio ha terminato (verificare che tale ordine è rispettato stampando i PID dei processi).

Esercizio 2

(15 min.)

Scrivere un programma per Windows/Unix che permette al processo principale **P** di creare un nuovo thread **T** il cui percorso di esecuzione è associato alla funzione “*thread_function*”.

Il processo principale **P** ed il nuovo thread **T** dovranno stampare ad output una stringa che li identifichi rispettando l'ordine $\mathbf{T} \rightarrow \mathbf{P}$, senza utilizzare “*WaitForSingleObject*”/“*pthread_join*”, ma sfruttando un concetto fondamentale che accomuna tutti i threads di un determinato processo.