

PRIMI ESEMPI DI OUTPUT AVENDO UN THREAD PADRE CHE CREA DUE THREAD FIGLI CHE ESEGUONO A STESSA START_ROUTINE

- 1) **dueThread.c** - poiché il main dopo avere creato i due figli, non aspetta i figli e fa una exit a livello di output potrebbe succedere che non si percepisca la creazione dei due thread figli!

```
osd@lica04:~/Pthread/primeProve$ dueThread
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
```

- 2) **dueThread-conReturn.c** – idem come sopra, dato che poiché il main dopo avere creato i due figli, non aspetta i figli e fa un return possiamo avere un output non corretto!

```
osd@lica04:~/Pthread/primeProve$ dueThread-conReturn
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
```

- 3) **Proviamo ora a cercare di far 'uscire' l'output dei due figli: dueThread-conSleep.c** –il main dopo avere creato i due figli, non aspetta i figli ma prima di eseguire la exit fa una sleep (di 10 secondi) e quindi otteniamo output corretto!

```
osd@lica04:~/Pthread/primeProve$ dueThread-conSleep
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread0 partito: Hello World!
Thread1 partito: Hello World!
```

- 4) **Proviamo ora a cercare di far 'uscire' l'output dei due figli, ma in un altro modo: dueThread-conScanf.c** –un risultato analogo a prima lo possiamo ottenere se il main dopo avere creato i due figli, non aspetta i figli ma prima di eseguire la exit fa una scanf e quindi si riesce a ottenere l'output corretto andando ad inserire il numero atteso solo dopo le stampe dei figli!

```
osd@lica04:~/Pthread/primeProve$ dueThread-conScanf
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Fornisci un valore intero al thread padre in modo da non morire prima dei thread figli!
Thread0 partito: Hello World!
Thread1 partito: Hello World!
10
Processo creatore ha letto 10
```

- 5) **dueThread-conPexitNelMain.c** –un risultato corretto lo possiamo avere se il main dopo avere creato i due figli, non aspetta i figli ma esegue (invece che exit o return) una pthread_exit.

```
osd@lica04:~/Pthread/primeProve$ dueThread-conPexitNelMain
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread0 partito: Hello World!
Thread1 partito: Hello World!
osd@lica04:~/Pthread/primeProve$ dueThread-conPexitNelMain
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread1 partito: Hello World!
```

Thread0 partito: Hello World!

6) dueThread-conJoin.c –un risultato corretto lo possiamo avere se il main dopo avere creato i due figli, li aspetta e solo dopo esegue una exit (o return).

```
osd@lica04:~/Pthread/primeProve$ dueThread-conJoin
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread0 partito: Hello World!
Thread1 partito: Hello World!
osd@lica04:~/Pthread/primeProve$ dueThread-conJoin
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread1 partito: Hello World!
Thread0 partito: Hello World!
```

7) dueThread-conJoinEPexitNeiThread.c –come prima, ma i figli tornano al padre un risultato (1000+id che è il loro numero d'ordine).

```
osd@lica04:~/Pthread/primeProve$ dueThread-conJoinEPexitNeiThread
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread0 partito: Hello World!
Thread1 partito: Hello World!
Pthread 0-esimo restituisce 1000
Pthread 1-esimo restituisce 1001
osd@lica04:~/Pthread/primeProve$ dueThread-conJoinEPexitNeiThread
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread1 partito: Hello World!
Thread0 partito: Hello World!
Pthread 0-esimo restituisce 1000
Pthread 1-esimo restituisce 1001
osd@lica04:~/Pthread/primeProve$ dueThread-conJoinEPexitNeiThread
Sto per creare il thread 0-esimo
Sto per creare il thread 1-esimo
Thread0 partito: Hello World!
Pthread 0-esimo restituisce 1000
Thread1 partito: Hello World!
Pthread 1-esimo restituisce 1001
```

8) dueThread-conJoinEPexitNeiThreadESelf.c –come prima, ma i figli oltre che l'indice di creazione stampano anche il loro tid (Thread Identifier) e lo stesso fa il padre.

```
osd@lica04:~/Pthread/primeProve$ dueThread-conJoinEPexitNeiThreadESelf
Sto per creare il thread 0-esimo
SONO IL MAIN e ho creato il Pthread 0-esimo con id=140431590430464
Sto per creare il thread 1-esimo
SONO IL MAIN e ho creato il Pthread 1-esimo con id=140431582037760
Thread1 partito: Hello World! Ho come identificatore 140431582037760
Thread0 partito: Hello World! Ho come identificatore 140431590430464
Pthread 0-esimo restituisce 1000
Pthread 1-esimo restituisce 1001
```