

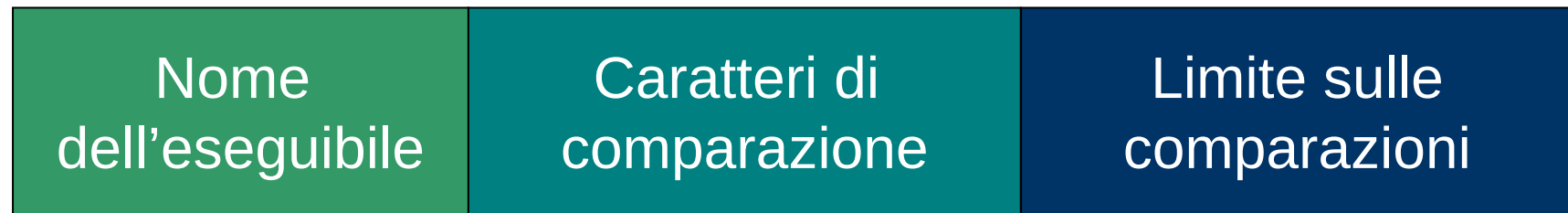
Esercitazione 4

segnali e handlers

Obiettivo 1/3

- Scrivere un comando che abbia la sintassi

compare <car1> <car2> <limit>



Obiettivo 2/3

- In particolare:
 - Il processo padre interagisce con l'utente chiedendogli di inserire una FRASE
 - Successivamente, il padre procede creando due processi figli (P1 e P2), mettendosi poi in pausa, in attesa dei loro segnali
 - P1 si mette in attesa
 - P2 ha il compito di effettuare comparazioni carattere per carattere tra la FRASE inserita con <car1> e <car2>
 - Se la comparazione tra il carattere corrente della frase e uno dei due parametri ha successo, P2 deve avvertire P1
 - P1 risponde all'avvertimento di P2 incrementando uno fra due contatori (e visualizzando lo stato dei contatori), **dipendentemente da quale comparazione ha avuto successo**

Obiettivo 3/3

- Normalmente, P1 si rimette in attesa
- Se però uno dei due contatori **sorpassa il <limite>** dato, P1 deve inviare un segnale al padre e poi terminare; il padre risponde al segnale terminando l'altro figlio (P2)
- Nel caso in cui P2 **consumi tutta la frase** data senza che nessuno dei due contatori sorpassi il limite, P2 provvede, prima di terminare, ad inviare un segnale al padre, che risponde terminando P1 (il quale è ancora in attesa...)
- Il padre dovrà essere avvertito mediante **un singolo segnale** (ad esempio **SIGUSR1**) e, in relazione a quale figlio glielo abbia inviato, provvederà a terminare l'altro figlio (e a raccoglierne lo stato) all'interno della procedura di gestione
 - Suggerimento: usare con criterio la primitiva **wait** e le funzioni messe a disposizione dalla libreria **sys/wait.h**

Problemi cruciali

- P1 deve comportarsi in maniera differente a seconda che P2 trovi un confronto di successo con <car1> o <car2> (o meglio, P2 deve avvertire P1 in due modi diversi)
- Il padre si comporta differentemente a seconda che il segnale arrivi da P1 o da P2 (infatti deve terminare l'altro figlio...)
- Il padre deve quindi capire, pur ricevendo sempre lo stesso segnale, quale dei due figli l'ha mandato

Esempio di esecuzione 1/2

```
[marco@localhost esercitazione4]$ ./compare a i 2
P0(3671): inserisci la frase
questa frase dice il falso
P1(3672): attendo P2...
P0(3671): creato processo P1 (3672)
P2(3673): inizio il ciclo di comparazione
P0(3671): creato processo P2 (3673)
P2(3673): comparazione di successo POSIZIONE=5 CARATTERE=a
P1(3672): contatore1=1, contatore2=0
P2(3673): comparazione di successo POSIZIONE=9 CARATTERE=a
P1(3672): contatore1=2, contatore2=0
P2(3673): comparazione di successo POSIZIONE=14 CARATTERE=i
P1(3672): contatore1=2, contatore2=1
P2(3673): comparazione di successo POSIZIONE=18 CARATTERE=i
P1(3672): contatore1=2, contatore2=2
P2(3673): comparazione di successo POSIZIONE=22 CARATTERE=a
P1(3672): contatore1=3, contatore2=2
P1(3672): limite superato
P0(3671): rilevazione terminazione volontaria del figlio 3672 con stato 0
P0(3671): ricezione del segnale da P1...termino P2
P0(3671): rilevazione terminazione involontaria del figlio 3673 a causa del segnale
SIGTERM
```

Esempio di esecuzione 2/2

```
[marco@localhost esercitazione4]$ ./compare a s 5
```

```
P0(3706): inserisci la frase
```

```
questa frase dice il falso
```

```
P1(3707): attendo P2...
```

```
P0(3706): creato processo P1 (3707)
```

```
P2(3708): inizio il ciclo di comparazione
```

```
P0(3706): creato processo P2 (3708)
```

```
P2(3708): comparazione di successo POSIZIONE=3 CARATTERE=s
```

```
P1(3707): contatore1=0, contatore2=1
```

```
P2(3708): comparazione di successo POSIZIONE=5 CARATTERE=a
```

```
P1(3707): contatore1=1, contatore2=1
```

```
P2(3708): comparazione di successo POSIZIONE=9 CARATTERE=a
```

```
P1(3707): contatore1=2, contatore2=1
```

```
P2(3708): comparazione di successo POSIZIONE=10 CARATTERE=s
```

```
P1(3707): contatore1=2, contatore2=2
```

```
P2(3708): comparazione di successo POSIZIONE=22 CARATTERE=a
```

```
P1(3707): contatore1=3, contatore2=2
```

```
P2(3708): comparazione di successo POSIZIONE=24 CARATTERE=s
```

```
P1(3707): contatore1=3, contatore2=3
```

```
P2(3708): frase consumata
```

```
P0(3706): rilevamento terminazione volontaria del figlio 3708 con stato 0
```

```
P0(3706): ricezione del segnale da P2...termino P1
```

```
P0(3706): rilevamento terminazione involontaria del figlio 3707 a causa del segnale  
SIGTERM
```