Esercitazione 11

Esercitazione di riepilogo

Agenda

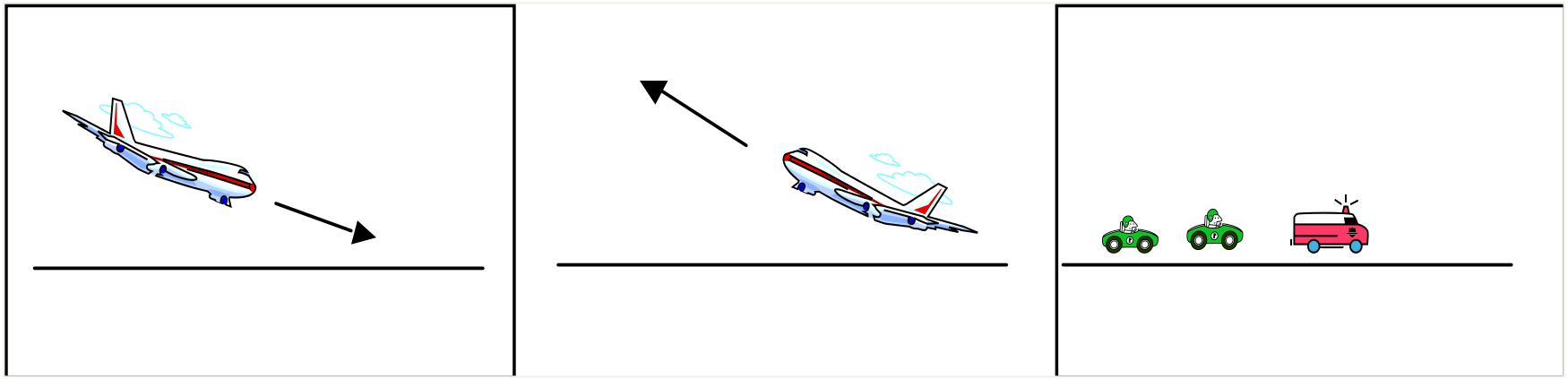
Esercizio 1 - Monitor java
Gestione della pista di un aereoporto

Esercizio 2 – Programma C di sistema
Gestione e monitoraggio dell'esecuzione del programma che risolve l'esercizio 1

Esercizio 3 – File comandi bash
Gestione e monitoraggio dell'esecuzione del programma che risolve l'esercizio 1

Esercizio 1

Sincronizzazione degli utilizzatori di una pista di atterraggio/decollo



Esercizio 1

Si consideri un piccolo Aeroporto nel quale è presente **una sola pista** utilizzata sia per i decolli che per gli atterraggi. La pista puo` anche essere percorsa da mezzi di servizio: **durante il transito di mezzi di servizio nessun aereo puo` accedere alla pista**.

Pertanto la pista puo` essere utilizzata in 3 modi diversi:

- **Atterraggio**: la pista e` allocata a un solo aereo che deve effettuare l'atterraggio.
- **Decollo**: la pista e` allocata a un solo aereo che deve eseguire il decollo.
- **Servizio**: la pista e` allocata a uno o piu` mezzi di servizio; per evitare situazioni di eccessivo traffico, in questo stato la pista non puo` accogliere piu` di MAX mezzi.

Esercizio 1 - Traccia (2/3)

Si vuole realizzare una politica di sincronizzazione degli accessi alla pista che tenga conto delle specifiche date e dei seguenti **vincoli di priorità:**

- nell'accesso alla pista venga data la **precedenza assoluta agli aerei in fase di atterraggio**;
- Inoltre, gli aerei in fase di **decollo abbiano la precedenza sui mezzi di servizio**.

Realizzare un'applicazione concorrente in Java che, rappresentando aerei e mezzi di servizio mediante thread concorrenti, implementi la politica di sincronizzazione data.

Esercizio 1 - Traccia (3/3)

In particolare, è richiesto che al termine di ogni decollo o atterraggio, l'aereo scriva sullo **standard** output, il tipo di operazione compiuta ("decollo" o "atterraggio"), il proprio ID e l'istante di tempo in cui si è conclusa l'operazione di decollo/atterraggio.

Per esempio:

- *"atterraggio AZ125 Fri May 31 12:16:12 CEST 2013"*

```
Date d=new Date(); //data corrente
d.toString() //dow mon dd hh:mm:ss zzz yyyy
```

http://docs.oracle.com/javase/6/docs/api/java/sql/Date.html

Esercizio sul monitor - Step

- Quali sono i thread?
- Qual è la risorsa condivisa?
- Quali variabili di stato ha?
- Quali sono le condizioni di sospensione?
- Quali vincoli di priorità? => Quindi ordine di priorità?

  - Lock
  - Condition
  - Contatori dei thread sospesi: scalari o array?

Esercizio 2

Programma C di sistema

Gestione e monitoraggio
dell'esecuzione del programma
che risolve l'esercizio 1

Esercizio 2 - Traccia (1/2)

Si realizzi un programma C di sistema che, utilizzando le system call Unix, preveda la seguente sintassi:

`lancia fileout timeout tipo_op`

Dove:

-**`fileout`** è il nome di un file scrivibile

-**`timeout`** è un intero positivo.

-**`tipo_op`** è un carattere che può assumere soltanto 2 valori: '**d**' o '**a**'.

Il processo iniziale P0 dovrà creare un processo figlio P1 il cui compito sarà quello di **eseguire l'applicazione java realizzata come soluzione dell'esercizio 1**, in modo tale che il suo output venga ridirezionato nel file **fileout**.

Esercizio 2 - Traccia (2/2)

Il processo P0, dopo **timeout** secondi dovrà terminare forzatamente l'esecuzione di P1.

Una volta terminato P1, P0 dovrà stampare sullo standard output il contenuto del file **fileout**, filtrato secondo il seguente criterio:

- **Se** `tipo_op = 'a'`, verranno stampate le linee contenenti la stringa "**atterraggio**".
- **Se** `tipo_op = 'd'`, verranno stampate le linee contenenti la stringa "**decollo**".

Esercizio 3

File comandi bash

Gestione e monitoraggio
dell'esecuzione del programma
che risolve l'esercizio 1

Esercizio 3 - Traccia

Si realizzi file comandi bash che preveda la seguente sintassi:

`lancia.sh fileout timeout tipo_op`

Dove:

-**`fileout`** è il nome di un file scrivibile

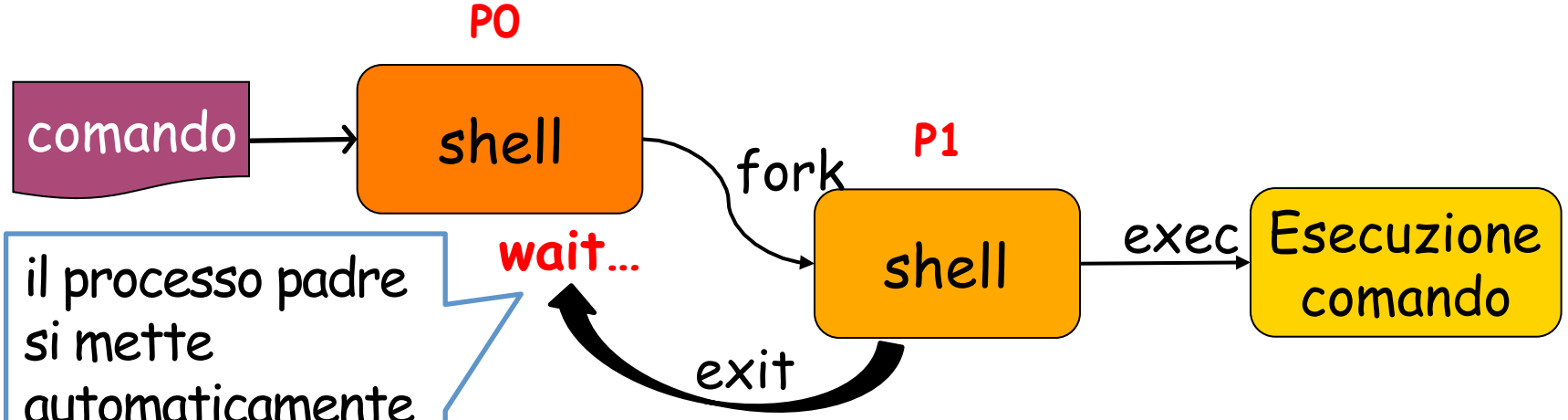
-**`timeout`** è un intero positivo.

-**`tipo_op`** è un carattere che può assumere soltanto 2 valori: '**`d`**' o '**`a`**'.

Il file comandi dovrà rispettare le stesse specifiche dell'esercizio 2.

Come genero un figlio P1 in bash?

- Quando la shell riceve un comando da eseguire:



il processo padre si mette automaticamente in attesa della terminazione del figlio.

Se al contrario NON voglio che il padre attenda la terminazione del figlio (perchè voglio che faccia qualcosa intanto):

```
-$ comando &
```

esecuzione in background!