



# procedi

capitolo 3 e 5 del libro (VII ed.)

# Processi

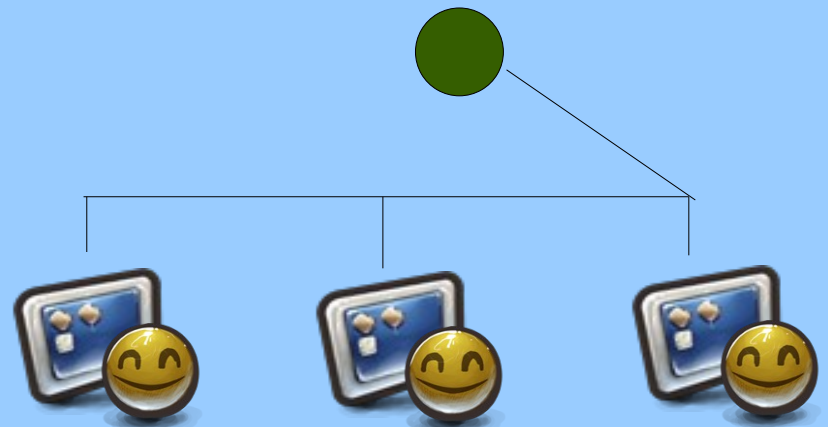


sistema batch



la CPU esegue i diversi job  
uno di seguito all'altro

il SO deve mantenere informazioni riguardo i diversi task: ogni task esegue un programma, elabora dei dati, ha un utente "proprietario", può avere una priorità. Un task viene **interrotto** e **ripreso**: occorre mantenere tutte le informazioni necessarie



sistema time-sharing:

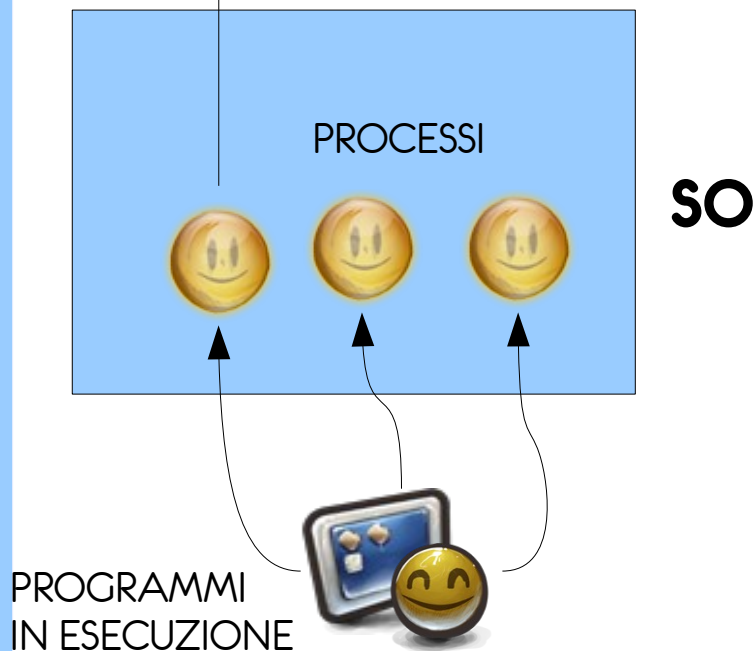
il tempo di CPU è diviso fra i task di più utenti collegati tramite terminali diversi

# Processi

- programma (o **sezione testo**)
- program counter (istruzione da eseguire)
- stack di esecuzione (con vrb, parametri, ind. di ritorno)
- heap (memoria allocata dinamicamente)

**SEZIONE  
DATI**

NB: ogni processo ha la propria sezione dati e non gli è consentito leggere o modificare le sezioni dati di altri processi!!!



“processo” è un’astrazione, una rappresentazione interna al SO, che consente di pensare e realizzare meccanismi quali multi-tasking, scheduling della CPU, protezione

# Esempio

Quanti processi erano in esecuzione sul mio portatile quando scrivevo queste slide?

slidesSisOp0607.odp - OpenOffice.org Impress

File Edit View Insert Format Tools Slide Show Window Help

Normal Outline Notes Handout Slide Sorter

1 2 3 4 5 6 7 8 9 10 11 12

processi

- programma (o sezione testo)
- program counter (istruzione da eseguire)
- stack di esecuzione (con vrb, parametri, ind. di ritorno)
- heap (memoria allocata dinamicamente)

SEZIONE DATI

NB: ogni processo ha la propria sezione dati e non gli è consentito leggere o modificare le sezioni dati di altri processi!!!

PROCESSI

SO

PROGRAMMI IN ESECUZIONE

"processo" è un'astrazione, una rappresentazione interna al SO, che consente di pensare e realizzare meccanismi quali multi-tasking, scheduling della CPU, protezione

Cristina Baroglio  
S.S. 2006/2007

Tas View x

Master Pages

Layouts

Custom Animation

Slide Transition

8.00 / 4.64 0.00 x 0.00 59% \* Slide 72 / 73 Default

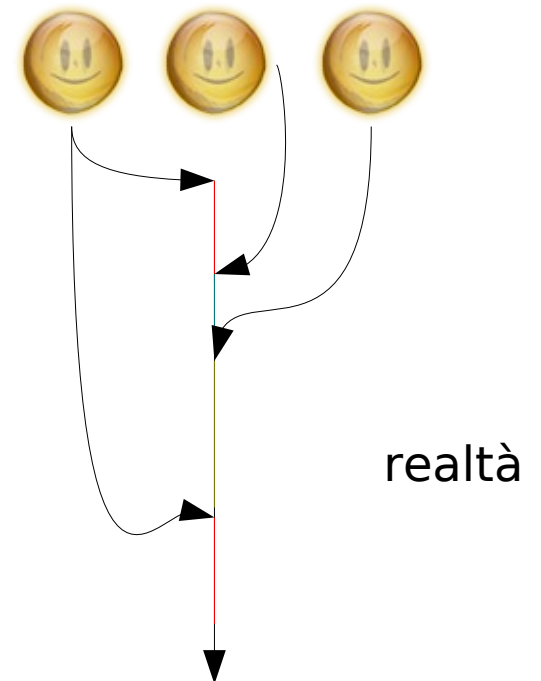
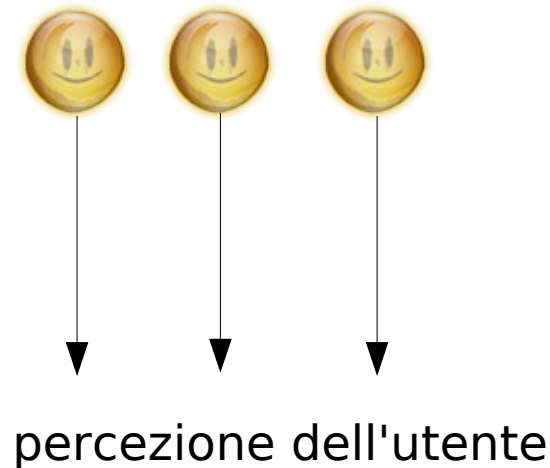
103

The image displays a stack of terminal windows, each titled "Shell No. 4 - Konsole". The foreground window shows the output of the `top` command, providing a snapshot of the system's current state. The output includes system load averages, uptime, and a list of running processes with their respective user, PID, PPID, CPU usage, memory usage, and command line.

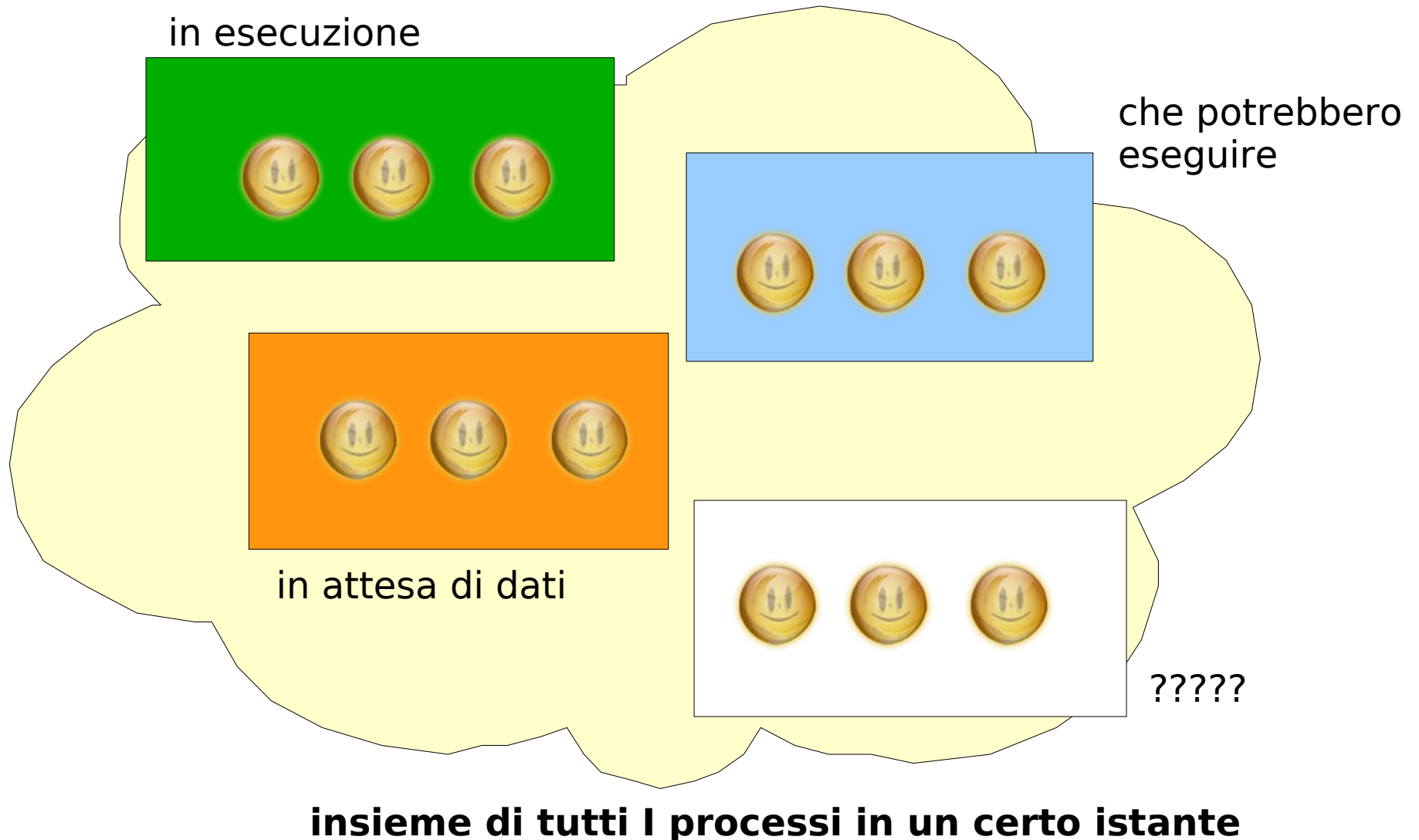
USER	PID	PPID	%CPU	%MEM	VSZ	RES	TIME	COMMAND
root	3390	0	0.0	0.0	11667	11812	0:00	skype -session 10138d4d9c9
root	3394	0	0.0	0.0	000116584	1093000	0:00	kdeinit
root	3408	0	0.0	0.0	3900	0	0:01	knotify [kdeinit]
root	3414	0	0.0	0.0	3911	0	0:00	/bin/bash
root	3435	0	0.0	0.0	3913	0	0:00	/bin/bash
root	3448	0	0.0	0.0	3915	0	0:00	/bin/bash
root	3464	0	0.0	0.0	3917	0	0:00	/bin/bash
root	3468	0	0.0	0.0	3947	0	0:00	kbluetoothd --dontforchesho
root	3487	0	0.0	0.0	3961	0	0:10	kpowersave [kdeinit]
root	3489	0	0.0	0.0	3969	0	0:01	knotify [kdeinit]
root	3535	0	0.0	0.0	4091	0	0:00	/bin/sh /usr/lib/openoffic
root	3553	0	0.0	0.0	4114	8.4	10:01	/usr/lib/openoffice/progra
root	3579	0	0.0	0.0	4162	0.0	0:00	kio_file [kdeinit] file /t
root	3604	0	0.0	0.0	4192	0.0	0:00	mp/ksocket-baroglio/klauncheru0Vo0b.slave-socket /tmp/ksocket-baroglio/kdefilepickerICyF4b.
root	3620	0	0.0	0.0	4251	0.0	0:00	[kacpid-work-1]
root	3661	0	0.0	0.0	21012	1.0	0:25	gqview
root	3686	0	0.0	0.0	29572	8.2	0:02	ksnapshot -caption KSnaphs
root	3687	0	0.0	0.0	29669	0.0	0:00	ps aux
root	3688	0	0.0	0.0	29670	0.0	0:00	more
root	3689	0	0.0	0.0	504	>>>		

# Parallelismo virtuale

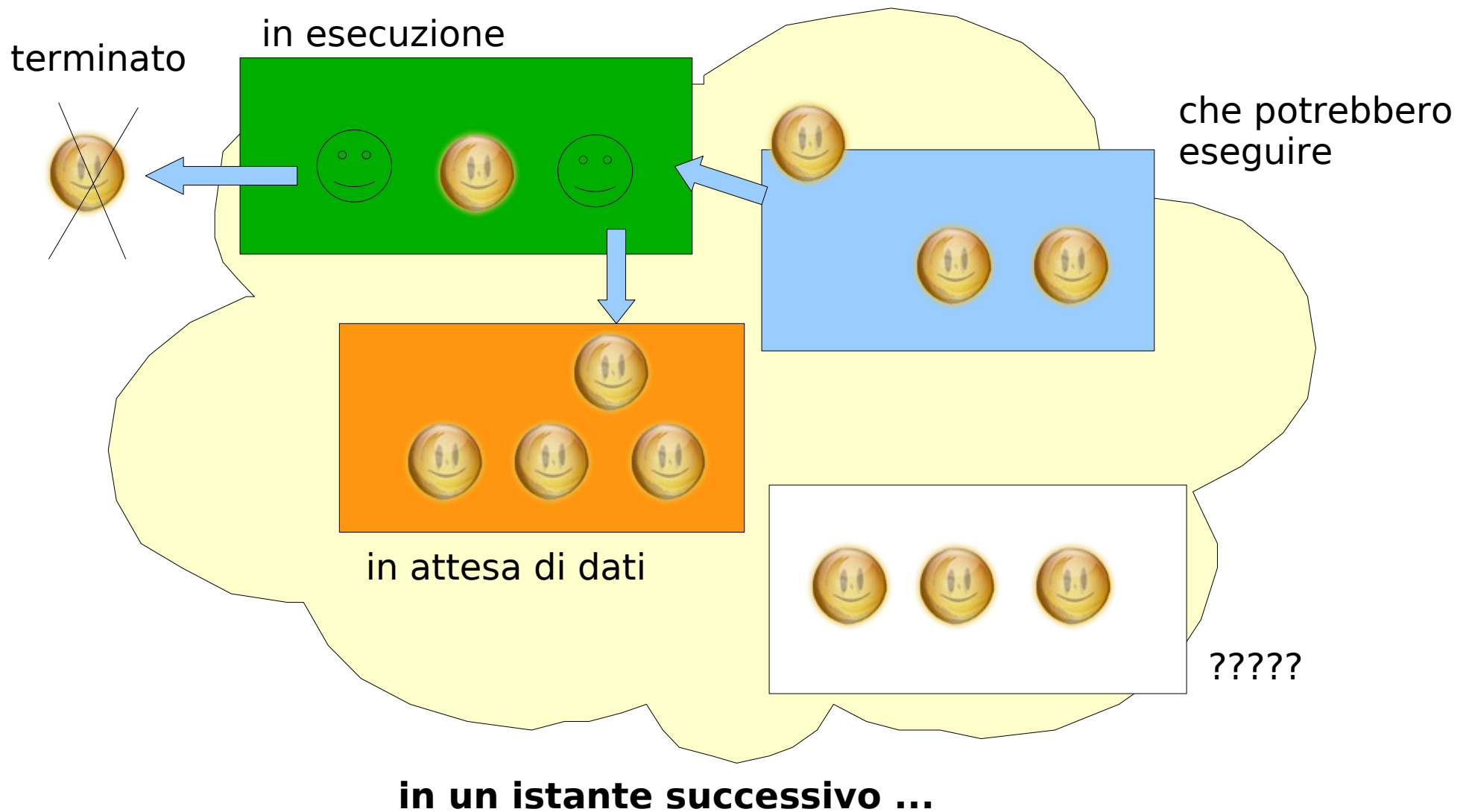
- Attraverso al SO i processi si suddividono l'uso delle risorse in modo tale portare avanti tutti quanti insieme la propria computazione
- Al più un processo per CPU può, in realtà, essere attivo in ogni istante ma gli utenti non se ne accorgono, percepiscono le diverse esecuzioni come parallele



# Categorie di processi



# I processi cambiano stato





# Parallelismo virtuale

- In un contesto in cui la CPU è una sola e i processi sono tanti (anche centinaia o migliaia) nasce l'esigenza di associare un'**informazione di stato** a ogni processo:
  - 1)**nuovo**, è lo stato di un processo appena creato
  - 2)**running**, in esecuzione
  - 3)**waiting**, in attesa di un evento (es. completamento di un'operazione di I/O)
  - 4)**ready**, il processo è pronto per essere eseguito ma al momento non ha assegnata la CPU
  - 5)**terminato**, ha cessato l'esecuzione

# Diagramma di transizione

- Diagramma di transizione degli stati di un processo

