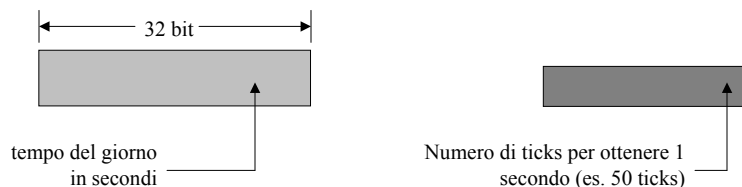


Gestione del temporizzatore

- Per consentire la modalità di servizio a *divisione di tempo* è necessario che il nucleo gestisca un *dispositivo temporizzatore* tramite un'apposita procedura che, *ad intervalli di tempo fissati*, provveda a sospendere il processo in esecuzione ed assegnare l'unità di elaborazione ad un altro processo
- **Gestione del clock:** i dispositivi *clock* generano interruzioni periodiche (*clock ticks*) a frequenze stabilite; la gestione software delle interruzioni consente di ottenere alcuni servizi quali:
 - aggiornamento della data
 - gestione del quanto di tempo (sistemi *time-sharing*)
 - valutazione dell'impegno della CPU di un processo
 - gestione della *system call* ALARM effettuata dai processi utente
 - gestione del *time-out* (*watchdog timers*)

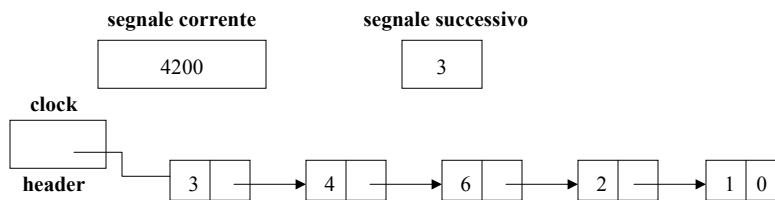
- **Aggiornamento della data** Il tempo del giorno viene mantenuto in secondi in un registro a 32 bit. Un contatore secondario conta i ticks (es.: frequenza 50 Hz) fino ad ottenere un secondo.



2^{32} secondi è circa 136 anni. Nei sistemi Unix il tempo si conta dal 1 gennaio 1970. Si avrà overflow del clock nel 2038 (l'interno a 32 bit è con segno).

- **Gestione del quanto di tempo.** Quando un processo inizia l'esecuzione, viene inizializzato un contatore con il valore del quanto di tempo espresso in clock ticks. Ad ogni interruzione il contatore è decrementato di 1. Quando raggiunge il valore zero, lo scheduler sottrae la CPU al processo.
- **Valutazione dell'impegno di CPU di un processo.** Ad ogni clock tick viene incrementato di 1 un campo contenuto nel descrittore del processo in esecuzione (problema delle interruzioni che possono avvenire durante l'esecuzione del processo).

- **System Call ALARM.** Un processo può richiedere al S.O. un segnale, un'interruzione, un messaggio, etc... dopo un certo tempo (es. pacchetto inviato sulla rete deve essere ritrasmesso, se non riconosciuto, entro un intervallo di tempo).



Nel caso siano attivi più segnali si simula la presenza di più clock tramite una lista ordinata. Ogni elemento della lista definisce quanti clock ticks occorre attendere per il prossimo segnale dopo il precedente. I segnali sono attesi a 4203, 4207, 4213, 4215, 4216.

Ad ogni tick il valore di "segnale successivo" (nell'esempio, 3) viene decrementato di 1. Quando diventa zero, viene generato il segnale corrispondente al primo elemento della lista. Questo viene rimosso dalla lista e "segnale successivo" viene aggiornato (nell'esempio, 4).

- **Time-out.** Può essere necessario fissare degli intervalli di tempo in cui attendere il verificarsi di un evento.

ES.: un processo attende una risposta da un altro processo per un tempo fissato.

Uso di un floppy disk: dopo l'accensione è necessario attendere 500 msec affinché raggiunga la velocità desiderata. Al completamento dell'I/O si attendono tre secondi (ad es.) prima di spegnere per verificare se arriva una seconda richiesta.

Differenza con il caso dei segnali: allo scadere del tempo, va in esecuzione una **procedura** (anziché inviare un segnale) che è parte del codice del chiamante. La procedura esegue un'azione prevista.