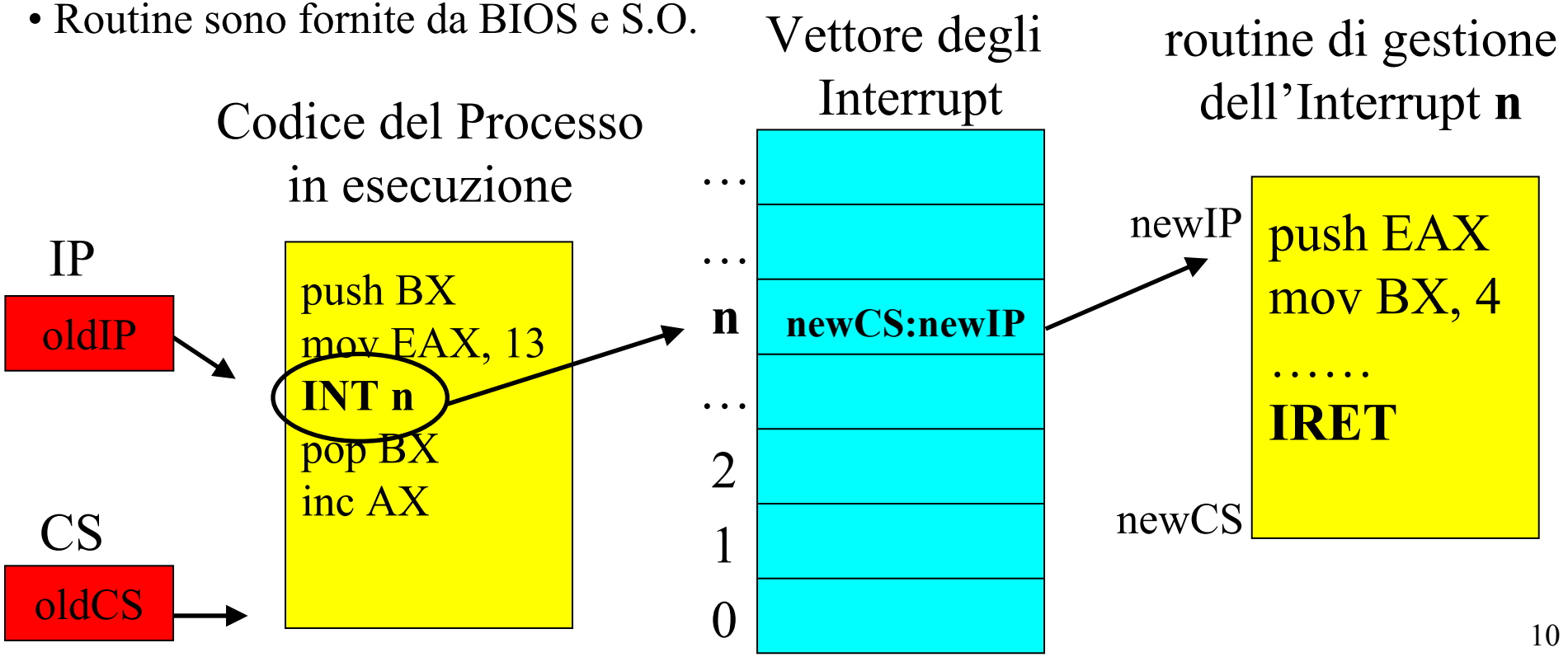
Chiamata ad Interrupt da Programma

- I programmi chiamano Interrupt per ottenere servizi da sistema operativo o BIOS.
- Per chiamare un interrupt da programma, si usa l'istruzione assembly: **INT n**
- n è un numero intero maggiore o uguale a zero che viene usato come indice per accedere alla n-esima posizione del Vettore degli Interrupt.
- Il Vettore degli Interrupt è posizionato a partire dall'indirizzo 0 in memoria.
- L' n-esimo elemento del Vettore di Interrupt contiene diverse informazioni, tra cui l'indirizzo (CS:IP) della prima istruzione eseguibile (codice macchina) della routine di gestione dell'interrupt di indice n.
- Routine sono fornite da BIOS e S.O.



Chiamata ad Interrupt da Programma – Cambio di Contesto

E' in esecuzione (1) il codice di un processo che sta utilizza lo stack utente.

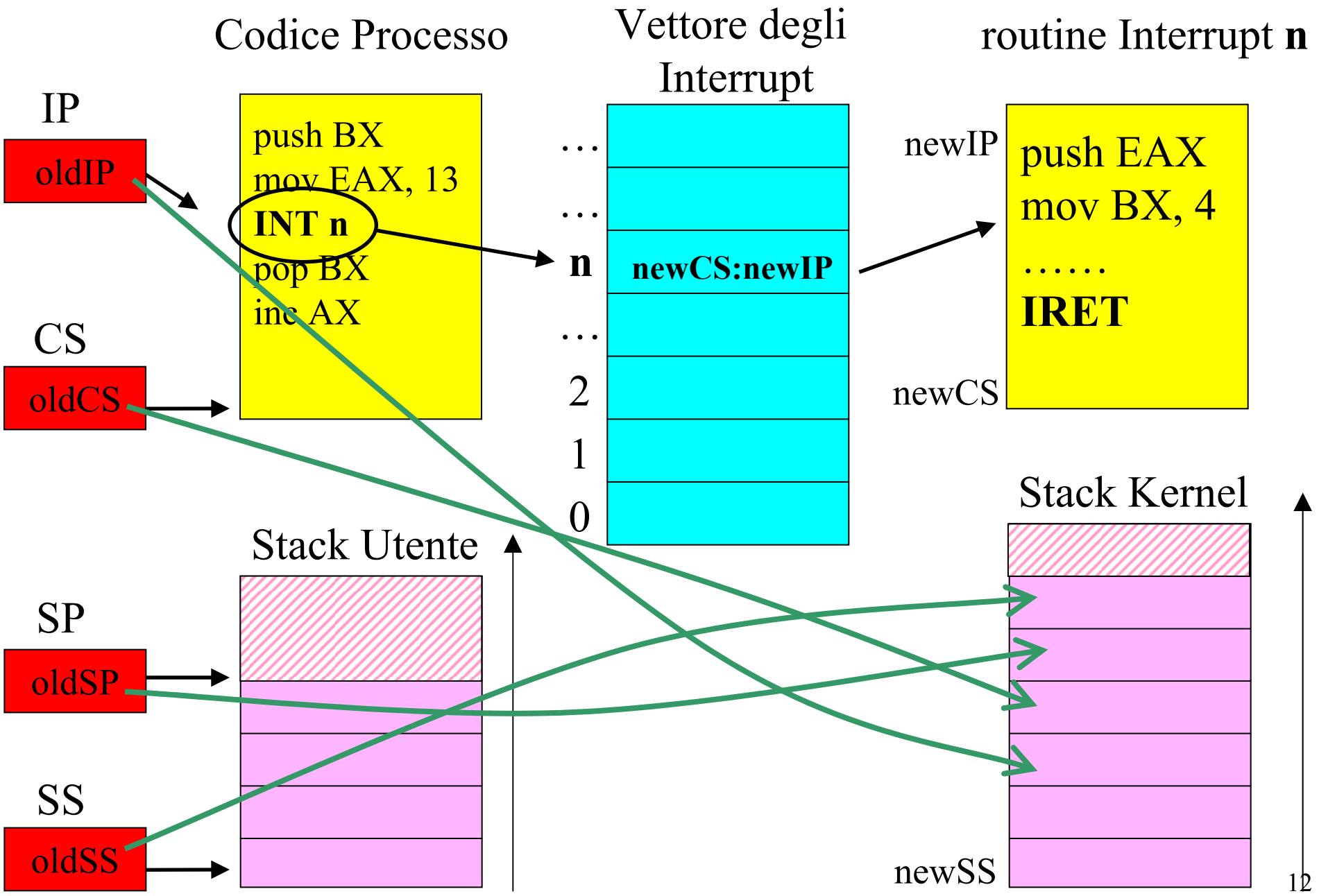
Quando la CPU esegue l'istruzione INT n :

- l' Instruction Pointer viene fatto puntare all'istruzione successiva ad INT.
- il processore passa in modalità kernel.
- il processore salva i valori attuali dei registri SS, SP, CS, IP nello stack di sistema del processo che ha chiamato l'interrupt, cioè lo stack dedicato all'esecuzione in modalità kernel per quel processo.
- il processore carica i registri SS e SP con i valori dello stack di sistema
- il processore carica i registri CS e IP con l'indirizzo della routine di gestione dell'interrupt trovato nell'n-esima posizione del vettore di interrupt.
- A questo punto (2) viene eseguita la routine dell'Interrupt, nello stack di sistema.

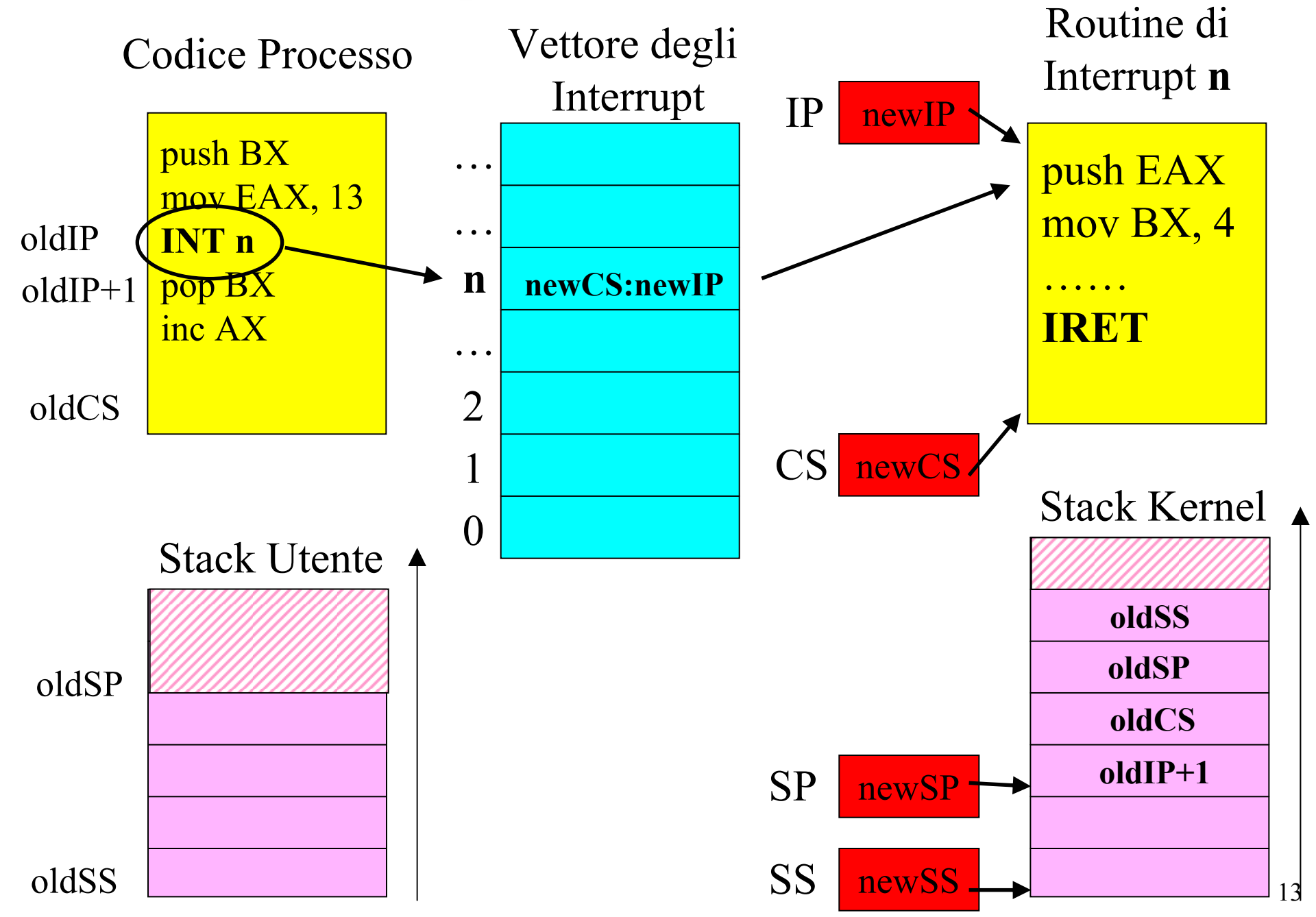
La routine dell'Interrupt termina (3) con l' istruzione IRET (Interrupt Return) che :

- carica i registri IP, CS, SP e SS con i valori salvati sullo stack di sistema.
- fa tornare il processore in modalità utente.
- così l'esecuzione ricomincia (4) dall'istruzione del programma successiva all'istruzione INT n utilizzando lo stack utente del processo.

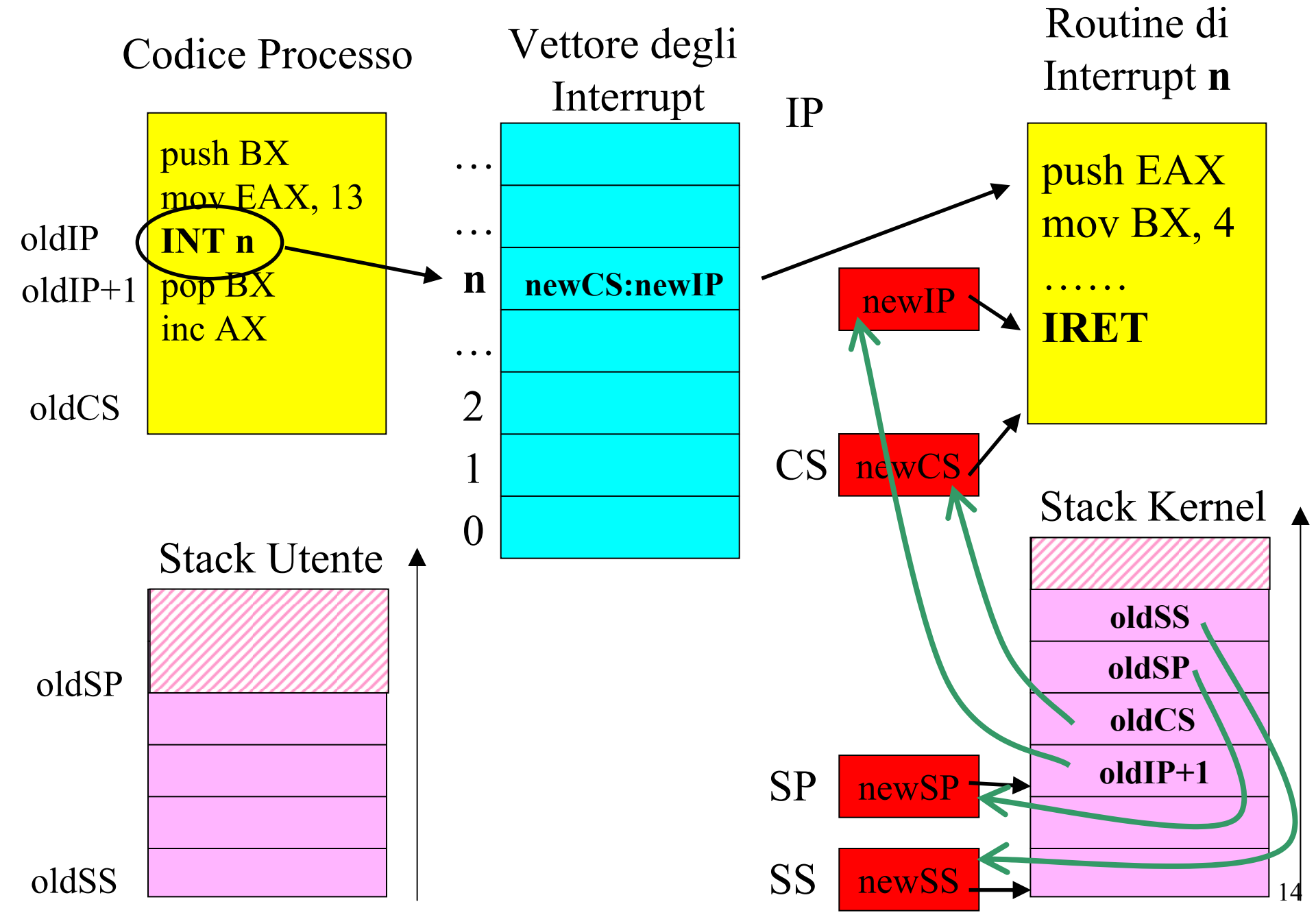
(1) Prima di Chiamata ad Interrupt da Programma



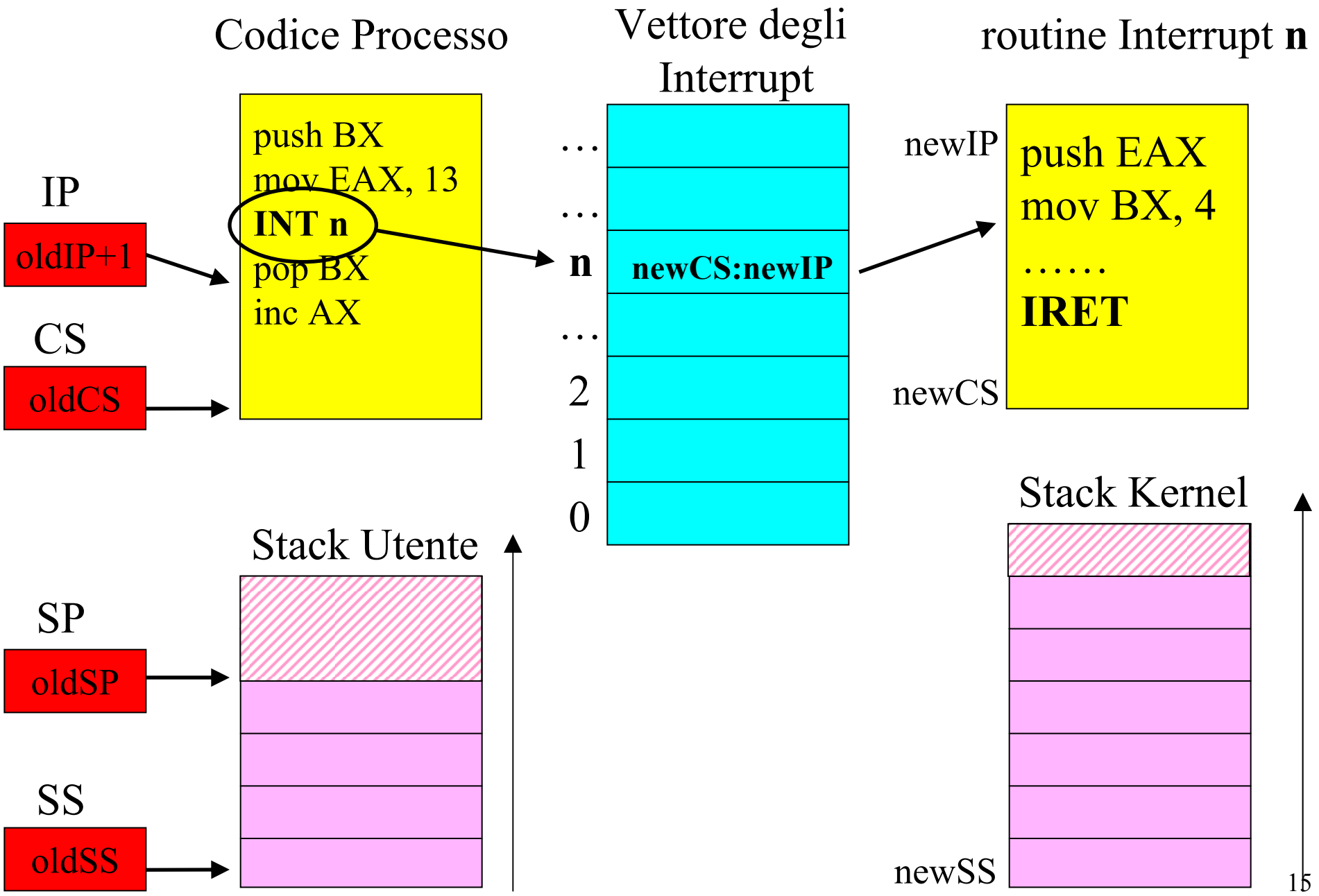
(2) Dopo esecuzione istr. INT n -



(3) Prima di esecuzione istr. IRET



(4) Dopo Esecuzione istr. IRET



Il meccanismo delle System Calls

- un programma utente, durante l'esecuzione puo' invocare una system call.
- la system call è implementata nella routine di gestione di un certo interrupt n.
- la system call viene invocata eseguendo una istruzione INT n, con un certo n intero
- il programma passa i parametri all' interrupt mettendoli in registri (EAX e altri)
- chiamata ad interrupt
- switching di contesto – da modo utente a modo kernel
    - salvataggio del contesto
- le routine di gestione degli interrupt.
    - puntatori a funzione.
- esecuzione su stack del kernel (perché?)
- ritorno a modo utente