

# Interfaccia dei processi (UNIX)

**fork** duplica il processo corrente, allocando un nuovo PCB e spazio di memoria.

Restituisce il PID del figlio nel padre, 0 nel figlio. La copia della memoria avviene con politica CoW, visto che spesso viene sostituita con una **exec**.

**execve** esegue un programma conservato sul disco, passando lista di parametri e ambiente. Carica il codice e copia i parametri, sostituendo il contenuto del processo che ha chiamato **execve**. Vengono mantenuti tutti gli altri attributi del processo (PID, padre, descrittori di file aperti...). Le altre **exec[vl]e?p?** sono funzioni di libreria.

**waitpid** mette in pausa il processo finché il figlio con il PID specificato (o nel gruppo specificato, o uno qualsiasi) non termina. Restituisce informazioni sul figlio: perché è terminato, l'eventuale codice di uscita, l'eventuale segnale che ha causato la terminazione. **wait** è una funzione di libreria.

**exit** termina il processo, salvando nel PCB l'exit code fornito. Il PCB non può essere liberato finché il padre non termina o chiama **wait** (il figlio esiste come *zombie*). Quando un processo termina i suoi figli (*orfani*) vengono adottati da **init** (PID 1).

**kill** invia un segnale ad un processo, o ad un gruppo di processi, o a tutti i processi. Processi non privilegiati possono mandare segnali solo a processi con lo stesso user ID.

**signal** imposta l'handler per un segnale. Il kernel fornisce handler di default per i processi che non li sostituiscono.