Funzioni di sistema per la gestione di processi nei s.o. UNIX/Linux
Corso di Sistemi Operativi IEL/IDT, a.a. 2002-2003
Ing. Jürgen Assfalg

```
pid_t fork()
```

1. creazione di un nuovo ingresso nella tabella dei processi, ovvero creazione di un nuovo Process Control Block (PCB); questo comporta l'assegnazione di un identificatore al nuovo processo, la registrazione dell'identificatore del padre, la definizione dello stato iniziale del nuovo processo—indicato con NEW oppure IDL (Image Definition and Loading, ovvero definizione dell'immagine di memoria e caricamento)—, nonché l'assegnazione delle informazioni necessarie per il calcolo della priorità. Per il resto si provvede a copiare i valori del padre;
2. se il codice è spartibile (ovvero, più processi possono fare riferimento allo stesso codice operativo) si procede all'aggiornamento della *tabella dei testi* (una tabella che tiene conto dei processi che fanno riferimento ad uno stesso segmento di codice in memoria), altrimenti si duplicherà anche il codice;
3. duplicazione dei segmenti dati e stack;
4. duplicazione dei dati di sistema, con qualche eccezione (gli indirizzi dei segmenti dati e stack saranno nuovi);
5. determinazione del valore di ritorno della funzione fork (il PID del figlio per il padre, e 0 per il figlio) e del valore del program counter, per il padre e per il figlio;
6. lo stato del figlio viene impostato a READY;
7. si riprende con l'esecuzione di un processo (quale esattamente, dipende dai criteri di scheduling implementati).

```
int execve(const char *filename, char *const argv [], char *const envp[]);
```

1. controllo sui diritti di esecuzione del programma specificato nella chiamata alla funzione;
2. viene rilasciata la memoria attualmente allocata al processo (memoria centrale, ma anche memoria la memoria su disco eventualmente utilizzata per lo swapping) e viene allocata la memoria necessaria alla nuova immagine del programma;
3. se il codice è spartibile, si verifica se è già presente in memoria ed eventualmente all'aggiornamento della tabella dei testi altrimenti lo si carica; se non è spartibile, si procede comunque al caricamento;
4. caricamento dell'area dati;
5. inizializzazione dei registri.

Si osservi che il programma invocato eredita il PID dal processo chiamante, così come tutti i descrittori di files aperti per i quali non è stato specificato che debbono essere chiusi a seguito di un'invocazione della funzione exec. In caso di successo la funzione non ritorna (l'esecuzione, infatti, passa al nuovo programma caricato), mentre restituisce il valore –1 in caso di errore.

```
void exit( int rv );
```

1. chiude i files (quindi, quando un processo termina naturalmente ma non ha chiuso tutti i files, il sistema operativo provvederà a chiuderli);
2. rilascio delle aree di memoria e del disco;
3. aggiornamento della tabella dei testi, ed eventuale rilascio della memoria assegnata al codice;
4. impostazione a ZOMBIE dello stato nella tabella dei processi (questo stato caratterizza un processo terminato, che ha liberato tutte le risorse, ma che viene tenuto nella tabella dei processi finché non viene rimosso);
5. se il processo non ha atteso la terminazione dell'esecuzione dei processi figli, questi vengono assegnati al processo di sistema *init*;
6. viene preparato il valore di ritorno;
7. si genera un segnale di terminazione e si notifica ai processi in attesa di un tale evento.

Nel caso di terminazione anormale di un processo, il sistema simula una chiamata alla funzione `exit()`, quindi liberando comunque le risorse allocate ad un processo e aggiornando le strutture dati necessarie al funzionamento del sistema operativo (tabella dei processi, tabella dei testi, ecc.).

```
pid_t wait(int *status);
```

1. scandisce la tabella dei processi per verificare se esistono processi figli nello stato ZOMBIE
   1.1. se esiste, allora vengono acquisite le informazioni di ritorno e viene definitivamente cancellato dalla tabella dei processi;
   1.2. altrimenti il processo si autosospende in attesa della terminazione dell'esecuzione di un suo processo figlio.