

## Funzioni EXEC()

Le funzioni della famiglia exec() servono per lanciare in esecuzione nuovi processi sostituendo l'immagine del processo chiamante con quella del processo chiamato

Il nuovo processo sostituisce completamente il vecchio programma

- Mantiene la stessa process structure (salvo le info relative al codice)
- Il PID del processo non cambia (stesso PID del padre)
- Ha codice, dati globali, stack e heap nuovi
- Mantiene le stesse risorse
- Mantiene le stesse variabili d'ambiente ( a meno dell'uso di execl o execlp)

Le funzioni della famiglia exec sono incluse nella libreria <unistd.h>, i prototipi sono:

```
int execl(char *pathname, char *arg0, ...,char *) NULL )
```

```
int execv(char *pathname, char *argv[])
```

```
int execlp(char *pathname, char *arg0, ..., char* envp[])
```

```
int execve(char *pathname, char *argv[], char* envp[])
```

```
int execlp(char *filename, char *arg0, ... )
```

```
int execvp(char *filename, char *argv[])
```

L'immagine del processo chiamante è sostituita con quella del processo indicato nel primo argomento. I parametri successivi consentono di specificare gli argomenti a linea di comando e l'ambiente ricevuti dal nuovo processo.

Queste funzioni ritornano solo in caso di errore, restituendo -1

Varianti dell'exec: a seconda del suffisso

- l : gli argomenti da passare al programma da caricare vengono specificati mediante una LISTA di parametri(terminata da NULL)
- p: il nome del file eseguibile specificato come argomento della system call viene ricercato nel PATH contenuto nell'ambiente del processo (es. execlp())
- v : gli argomenti da passare al programma da caricare vengono specificati mediante un VETTORE di parametri(es. execv())
- e: la system call riceve anche un vettore (envp[]) che rimpiazza l'environment (path, direttorio corrente, etc.) del processo chiamante (es. execve())

la execl() normalmente si utilizza quando il numero di argomenti è noto a priori mentre la execv() quando il numero degli argomenti può variare dinamicamente. Normalmente argv contiene i parametri.

Esempio: uso execve()

```
int execve(char *pathname, char *argv[], ..char * envp[]);
```

pathname è il nome (assoluto o relativo) dell'eseguibile da caricare

argv è il vettore degli argomenti del programma da eseguire

env è il vettore delle variabili di ambiente da sostituire all'ambiente del processo (contiene stringhe del tipo "VARIABILE=valore")

```
char *envp[]={ "USER=4IB", "PATH=/home/4IB/TeP", (char *)0};
char *argv[]={ "ls", "-l", "lista", (char *)0};
main()
{int pid, status;
pid=fork();
if (pid==0)
{execve("/bin/ls", argv, envp);
cout<<"exec fallita!\n";
exit(1);
}
else if (pid >0)
{ /* processo padre..... */
}
```