

Sistemas Operativos

Tema 2 Administración de Procesos

Ing. Rafael Sandoval Vázquez.

Llamada al sistema fork()

La llamada al sistema fork() es usada para crear procesos. Esta no toma argumentos y regresa el ID del proceso creado. El propósito de fork() es crear un nuevo proceso, donde este se convierte en el hijo del proceso llamador.

Llamada al sistema fork()

Después de creado el nuevo proceso “hijo”, ambos procesos ejecutarán la siguiente instrucción que se encuentra después de fork(), por lo tanto es necesario distinguir el padre del hijo, esto se puede comparando el valor devuelto por fork().

Llamada al sistema fork()

Si fork() regresa un valor negativo, el nuevo proceso no se pudo crear.

Si fork() regresa cero podemos reconocer al hijo

Si fork() regresa un valor positivo, el PID del proceso hijo, se puede distinguir al proceso padre.

Llamada al sistema fork()

Al momento de la creación de un nuevo proceso, UNIX hace una copia exacta del espacio de direcciones del proceso padre y esta copia sirve para crear al proceso hijo. Ahora el proceso padre e hijo tendrán un espacio de direcciones separado.

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#define MAX_COUNT 200
#define BUF_SIZE 100
void main(void) {
    pid_t pid;
    int i;
    char buf[BUF_SIZE];
    fork();
    pid = getpid();
    for (i = 1; i <= MAX_COUNT; i++) {
        sprintf(buf, "Este es el proceso %d, valor = %d\n", pid, i);
        write(1, buf, strlen(buf));
    }
}
```

Parent

```
main()
{
    pid = 3456
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void ChildProcess()
{
    .....
}

void ParentProcess()
{
    .....
}
```

Child

```
main()
{
    pid = 0
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void ChildProcess()
{
    .....
}

void ParentProcess()
{
    .....
}
```

Parent

```
main()    pid = 3456
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

void ChildProcess()
{
    .....
}

→ void ParentProcess()
{
    .....
}
```

Child

```
main()    pid = 0
{
    pid=fork();
    if (pid == 0)
        ChildProcess();
    else
        ParentProcess();
}

→ void ChildProcess()
{
    .....
}

void ParentProcess()
{
    .....
}
```


Llamada al sistema wait()

- Esta función suspende la ejecución del proceso actual hasta que alguno de sus procesos hijos haya terminado o reciba alguna otra señal.
- wait() almacena la información de estado del proceso llamador en la dirección de memoria que le es pasada como argumento y regresa el PID del proceso completado.