```
75    case EXEC:
76      ecmd = (struct execcmd*)cmd;
77      if(ecmd->argv[0] == 0)
78        exit(1);
79      exec(ecmd->argv[0], ecmd->argv);
80      fprintf(2, "exec %s failed\n", ecmd->argv[0]);
81      break;
```

In the sh.c(actually it's shell) we have case EXEC and when we press ENTER, we input "\n", not null, actually. Then the process would execute exec in the case. And when we execute exec, we would create a new process which cause the pid in table increase 1. There is a pagetable used in exec.c in kernel space and every process have their own PCB which contains their pid、page and so on.

```
int
exec(char *path, char **argv)
{
  char *s, *last;
  int i, off;
  uint64 argc, sz = 0, sp, ustack[MAXARG], stackbase;
  struct elfhdr elf;
  struct inode *ip;
  struct proghdr ph;
  pagetable_t pagetable = 0, oldpagetable;
  struct proc *p = myproc();
```