10. XV6 has a mechanism to take an existing user process and fork a new user process This new process can continue executing the code or call exec to load a new application off ~~of~~ disk and run it. However before init is started, There are no user processes at all; init is this very first user process

The PPID of every logic shell is always

1. This is the init process: the second process of system.

- Init is very important process and apart from being parent of users shells, it is also responsible for giving birth to every service that's running in the system like printing, mail

<u>algorithm</u> : <u>init</u>

input - none
output - none

```
    fd = open ("etc/initlab", O_RDONLY);
    while ( line = read (fd, buffer) )
    {    if ( invoked state ! , buffer state )
            continue;
        if ( fork () = = 0 )
        {    execl ( "process specified in buffer");
             exit ();
        }
    while ((pid == wait((int *) 0)) != -1)
    {
        check here if a spawned child died;
        consider respawning it
        otherwise, just continue;
    }
  }
}
```

System calls for

time (tloc)

where tloc points to a location in user process for the return value, time returns this value from system call, too.

times

retrieves the cummulative times that the calling process spent executing in user mode and kernel mode and the cumulative times that all the zombie children had executed in user mode and kernel mode.

times () stores current process times in struct tms.

```
struct tms {
    clock_t   tms_Utime;
    clock_t   tms_stime;
    clock_t   tms_cutime;
    clock_t   tms_cstimes;
}
```