non-Strict in create   use GNU extensions

for UID and GID   write the number

| 80 |

if header->uid[0] == 0x80

Process Lifecycle:

Birth: fork()

Life: does process thing

Death: termination

↓ Afterlife: waiting

normal:

- return from main
- call exit()
- call _exit()

not normal

- call abort()
- uncaught signal

```
int main(void){
  pid_t child;
  if((child=fork())){
    /* parent */
    if(child == -1){
      perror("fork");
      exit(EXIT_FAILURE);
    }
    printf("Hi, kid\n");
  }else{ /* child */
    printf("Hi, mom\n");
  }
  return 0;
}
```

↑

Zombie

pid-t fork(void):

↳ creates duplicate of current process

returns -1 on error

pid of child in parent

Ø in child

pid_t wait(int *status)

↳ Block caller until a child terminates

Returns:

-1 on error

pid of terminated process on success

if status isn't NULL → encodes cause of death & exit status

WIFEXITED(status)

WEXITSTATUS(status)

WIFSIGNALED(status)

WTERMSIG(status)

Loading a new program: the execs

int execl (const char *path, const char *argv0, ...,)

Returns

-1 on error

environment: NAME =value