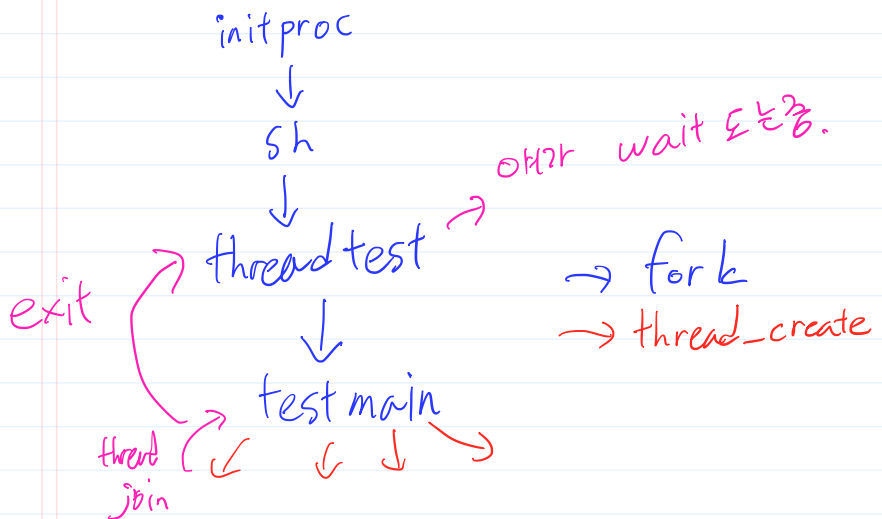


	int	tid
struct proc	proc*	mthread
	proc*	cthread [NPROC]
	void*	ret [NPROC]
	jint	vstack [NPROC]

52 라 변경 되면 항상 모든 thread 변경하는 것.



allocproc

전부 초기화
mthread = corproc
cthread, ret, vstack 전부 NULL
rrlast = 0.

fork

vstack NPROC 만큼 전부 설정.
cthred[0], vstack[0] 이 처음
tid = pid로 설정. parent = mproc
!!! 스레드가 돌고있는 경우 예외 처리하기!!!

thread_create

tid = allocproc에서 받은 pid
pid = mproc → pid.
pgdir = mproc → pgdir
cthred 0만큼 받아서 vstack, cthread set.
mthread → percentage == 0
init mlfq

scheduler

set_cpu_share call하면 스레드 전부 pop.
MLFQ는 2가지
stride는 main thread만.
내부에서 round robin 돌리기.

thread_exit

main thread가 끝나면 `exit()`으로
파일 닫고.

i를 찾아서 `mproc->ret[i]`에 저장.

`state` → ZOMBIE.

`wakeup (mproc)`

`sched()`

Thread-join

복합 프로그램 안에서

1. `cthread->tid == thread-t` 찾기.
→ 못찾으면 리턴 -1

2. `state == ZOMBIE`
retval 저장. / `kfree(kstack)`

mthread의 cthread 죽이기 포함

상태들 초기화.

`state == UNUSED` 세팅.

`mproc->killed`하면
return 1.

exit

일단 cthread 정리가 필요함

cthread 돌면서 있으면.

파일 닫고
기존 exit 프로그램만 해주고.
ZOMBIE로 변경.

mproc에서

파일 닫고

ZOMBIE로 바꾸고.

children 다 남기고.
wake up (mproc → parent)

exec

누가 콜하든 mproc 데이터 전송처리.

남기고 전송 종료만 듣고 parent 깨우고.

나 자식을 mthread로 변경.

pgdir, ostack, cthread 등 프로그램

kill

종료된 후 다시 프로세스를 mproc에다

kill 시킨다.

kill 체크는 mproc을 포함할 것.

pipe

sbrc

조건에 따라 2개.

이 경우 groupproc (이것) 실행.

group proc 2개

