MiniOS pthread改动以及测试的规范化

* + 1. v0.1 董章祺

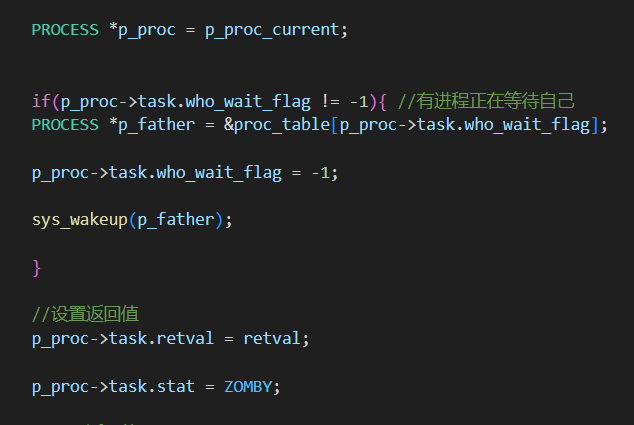
1. 改动简介

1)MiniOS中原有pthread\_create()系统调用，但没有pthread\_exit()和pthread\_join()，本次改动增加了这两个系统调用。

2)编写了一些系统调用测试文件，并对系统调用测试进行规范化。

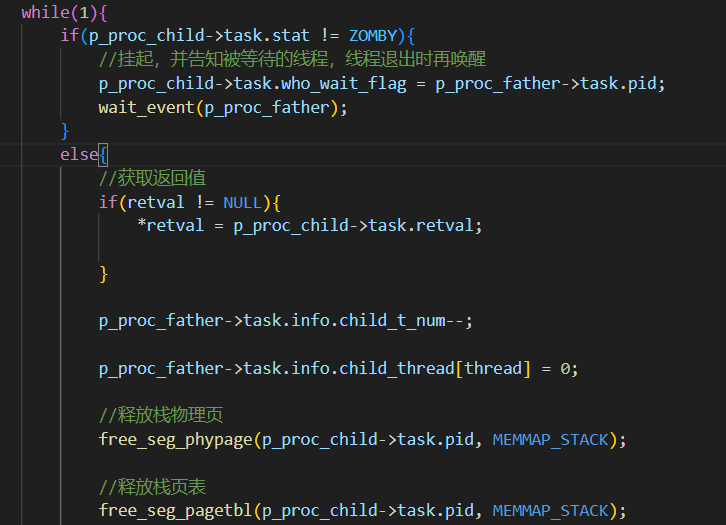
1. 详细介绍
2. pthread\_exit()及pthread\_join()

调用pthread\_exit()为一个线程的退出。线程调用pthread\_exit()后，检查who\_wait\_flag确认是否有进程在等待自己(可能有进程调用pthread\_join()在等待自己而挂起)，如果有就唤醒对应进程。然后设置返回值，修改线程状态。



pthread\_exit()核心代码

调用pthread\_join()等待目标线程退出。进程调用pthread\_join后，检查要等待的线程的状态，如果不是ZOMBY则代表目标线程未退出，此时先将自己挂起，并告知目标线程。如果是ZOMBY则获取pthread\_exit传递的值，然后回收目标线程的资源。

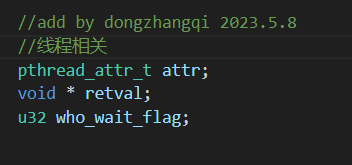


pthread\_join()核心代码

注意，Linux中，线程属性中detachstate为PTHREAD\_CREATE\_JOINABLE的线程在退出时不会主动释放资源，只有在其他进程或线程显式的调用pthread\_join后才会释放资源，而PTHREAD\_CREATE\_DETACHED的线程退出时会由系统自动回收资源。目前MiniOS中，所有进程默认都是JOINABLE的(与Linux的缺省情况相同)。

2)pthread\_exit()及pthread\_join()涉及到的改动

①在PCB中添加了三个变量，分别用于线程属性、记录exit返回值以及记录在等待的线程。

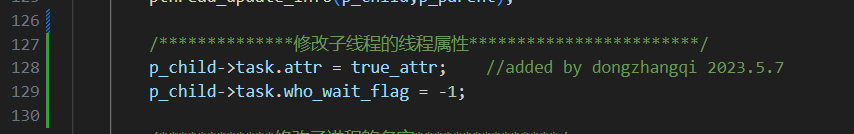


②修改了PCB中的proc\_stat，把冗余的wait\_exit\_flag给去除了，合并到proc\_stat中，修改了全部涉及wait\_exit\_flag的代码，并顺便解决了wait和exit中存在的忙等待问题。



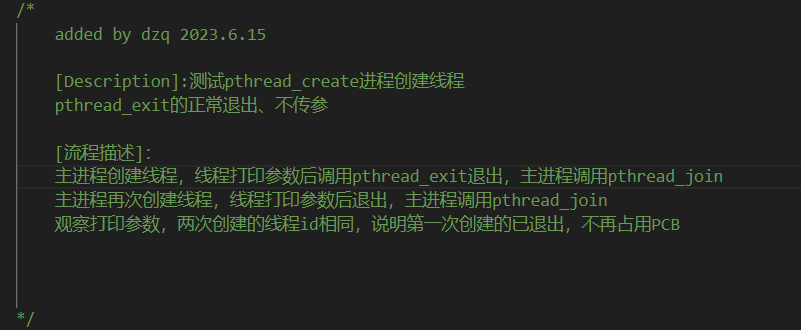
1. 修改了一些pthread\_create的代码。





3)系统调用测试的规范化。

1. 命名：t\_\*xx 例如t\_pthr02。t代表这是测试文件，pthr描述该测试的主体，02是序号。
2. 描述：为了方便测试文件的理解，请撰写对测试文件的描述，包含撰写人、撰写时间、测试什么系统调用的什么功能，以及代码流程的描述。如下图示例：



1. 每个文件只测试一种情况，如果要测试其他情况请另建一个文件，而不要通过注释代码来在同一个测试文件里进行修改，以便保留更多的测试用例供回归测试使用。