به نام خدا

گزارش پروژه دوم xv6

محمد جواد زندیه 9831032

در این گزارش فایل هایی که تغییر داده شده اند به همراه بخش هایی که تغییر یافته است را بررسی می کنیم:

1. Proc.c

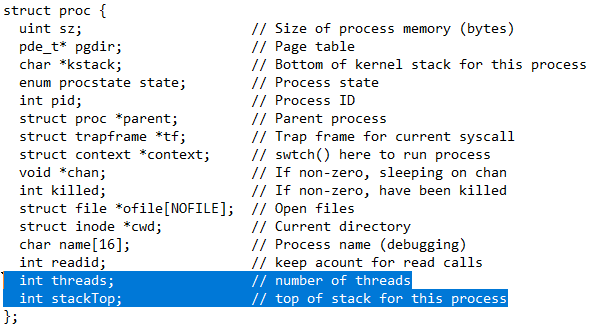
برای انجام کارهای مختلف نیاز است تا به Race condition هم دقت شود و به همین منظور برای ترد هایمان یک قفل تعریف می کنیم.





در هنگام allocate کردن یک پردازه باید به thread, stackTop هم مقدار بدهیم زیرا در فایل proc.h و در PCB مربوط به پردازه ها این دو ویژگی جدید را اضافه کرده ایم، مقدار دهی اولیه می کنیم:

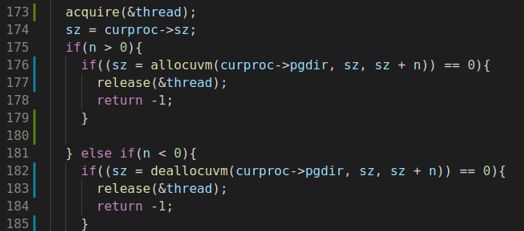


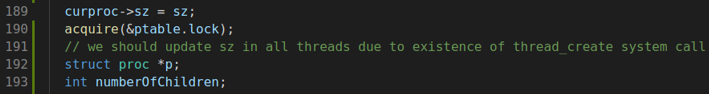


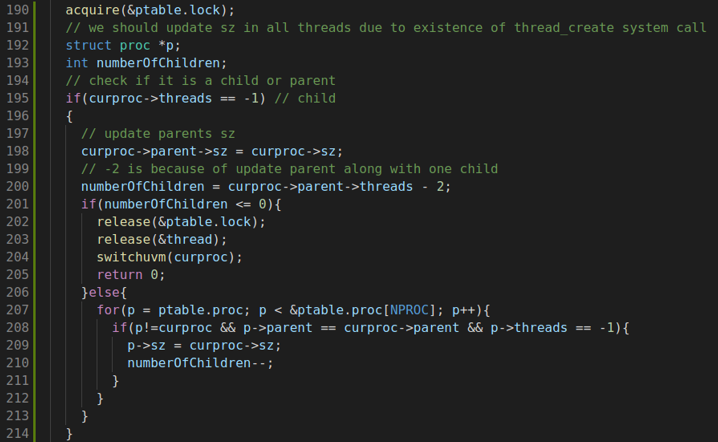
وقتی که یک پردازه را ایجاد می کنیم در ابتدا فقط خودش به عنوان ترد این پردازه می باشد:

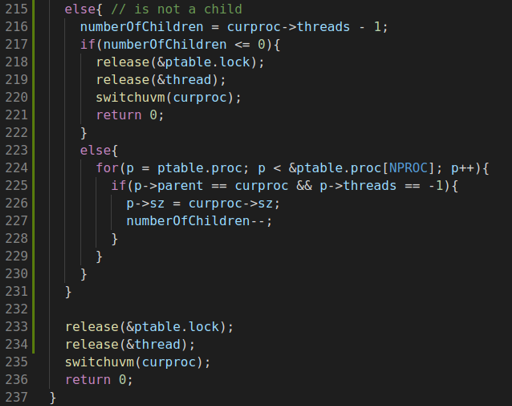


در هنگام گسترش فضای آدرس یک پردازه باید فضای آدرس ترد های آن نیز افزایش یابد در صورت وجود چون فضای ترد و پدرش یکسان است.

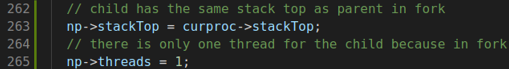




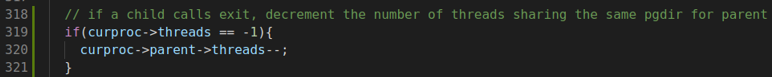


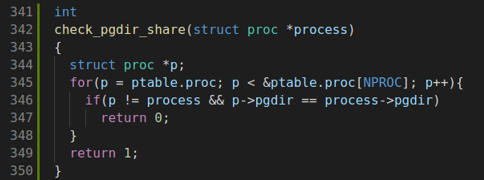


ترد ها دارای stackTop یکسان با پدر خود هستند و وقتی fork می کنیم یک ترد اضافه می شود پس:



اگر تردی Exit را صدا بزند نباید بلافاصله فضای آدرس را free کنیم چون ممکن است ترد های دیگری در حال استفاده از این فضای آدرس باشند پس باید دقت شود که آیا تردی دیگری در حال استفاده از آن هست یا نه





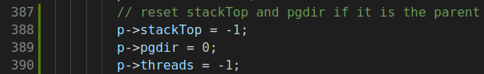
Thread\_wait روی ترد های فرزند عمل خواهد کرد در wait باید اگر ترد فرزند نیست wait کنیم و بالعکس پس:



همچنین برای free کردن هم می بایست چک شود که آخرین ترد باشیم:



چون در تابع wait مربوط به پردازه هستیم و نه ترد پس باید threads آنرا -1 قرار دهیم

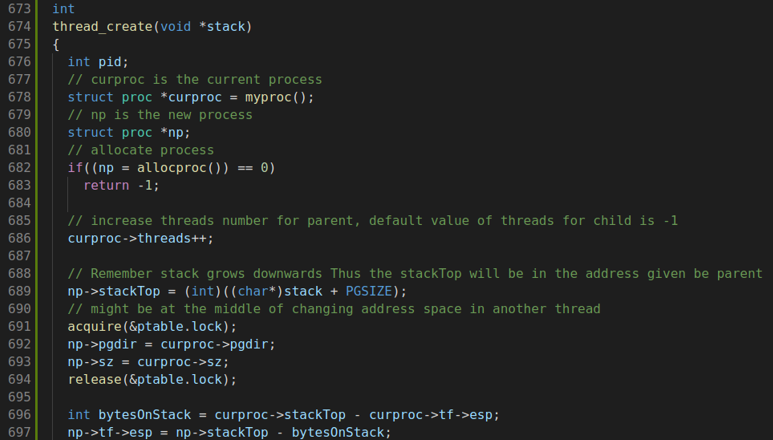


حال به پیاده سازی thread\_create می رسیم:

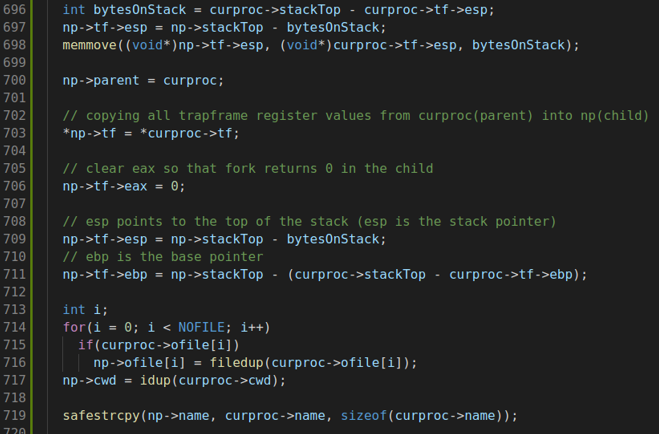
ابتدا پردازه کنونی که myproc هست را پیدا کرده و یک پردازه جدید از آن allocate می کنیم

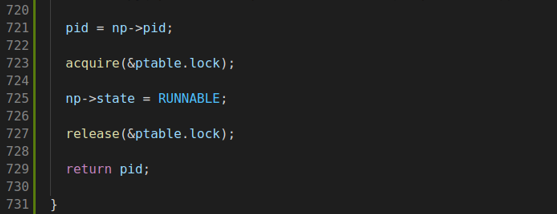
سپس تعداد ترد های پردازه curproc را یک واحد افزایش می دهیم چون قرار است پردازه جدید به عنوان ترد این پردازه باشد

سپس stackTop پردازه و ترد آنرا یکسان کرده و فضای هر دو را یکسان می کنیم

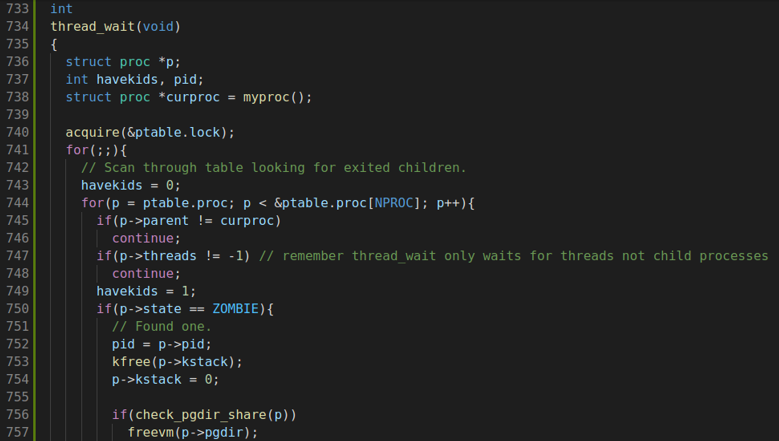


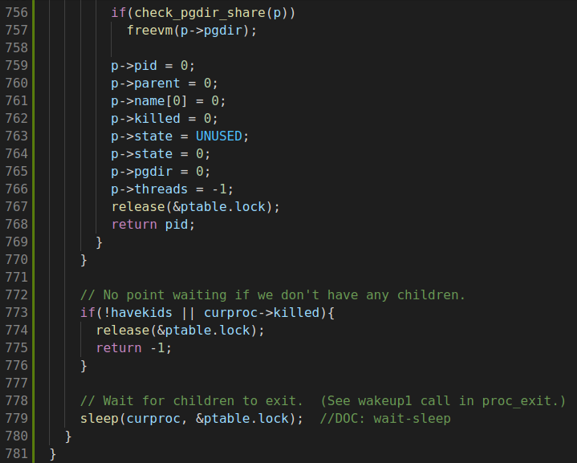
تعداد بایت هایی که در استک باید ذخیره شود را بدست می آوریم و تمام آنرا از پدر در ترد فرزند کپی می کنیم





حال در thread\_wait هم داریم که:





باید دقت شود که در اینجا فقط برای ترد ها wait می کنیم و نه پردازه ها

1. حال باید دو سیستم کال جدید را در سیستم عامل تعریف کنیم و مشابه فاز یک عمل می کنیم:

Syscall.c





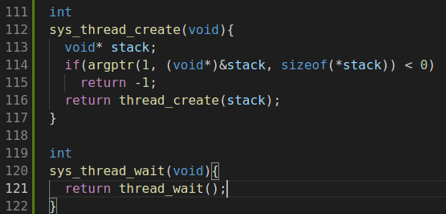
Defs.h



Syscall.h



Sysproc.c



User.h

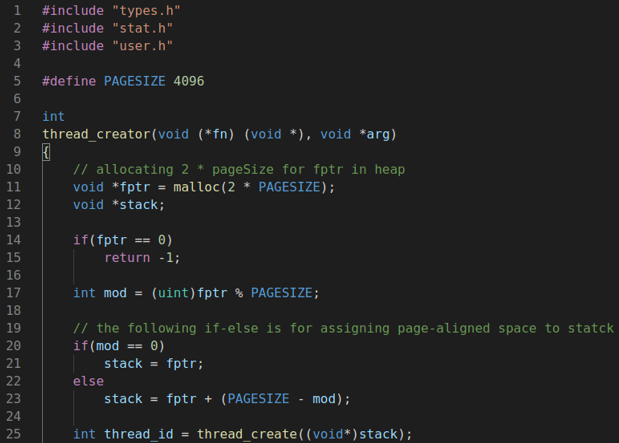


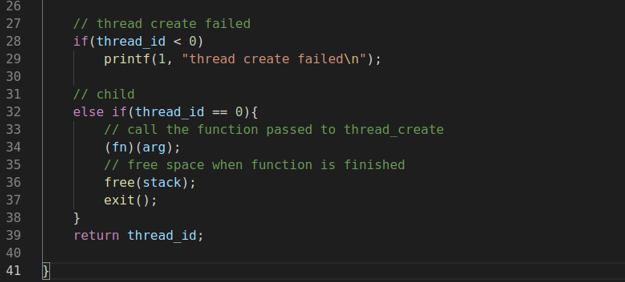
Usys.S



1. حال باید یک تابع پوششی به عنوان thread\_creator بسازیم:

در این تابع یک ترد جدید ایجاد می شود و آی دی آن باز گردانده می شود.





این تابع را باید در Makefile و user.h معرفی کنیم:





1. همچنین باید stackTop را در فایل exec.c مقداری دهی اولیه کنیم



1. سپس باید تست بنویسیم و در Makefile هم این تست ها را بشناسانیم به xv6



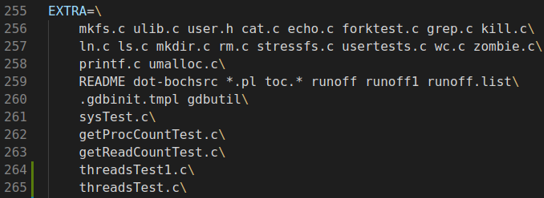
این تست بالا برای تست کردن درسی عملکرد تابع پوششی thread\_creator می باشد

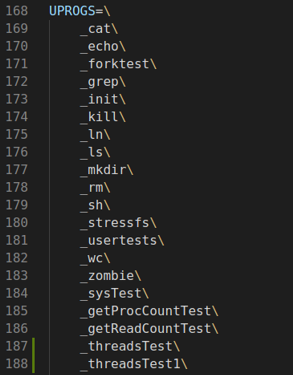
در تست زیر هم داریم:





حال در Makefile تعریف می کنیم تست هایمان را:





1. در نهایت خروجی:



