پروژه سیستم عامل - فاز 1

امیرعلی فرازمند

احسان احمدپور

فهرست:

* فایل های تغییر داده شده و اضافه شده
* کد های موجود در هر فایل تغییر داده شده و اضافه شده
* توضیح کلی و منابع
* خروجی برنامه

فایل های تغییر داده شده:

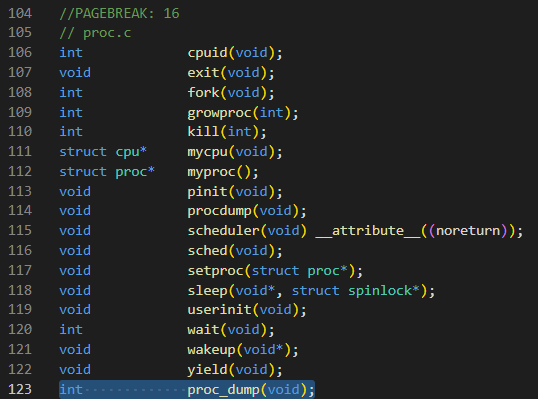
* defs.h
* syscall.h
* proc.c
* Usys.s
* Makefile
* proc.h
* syscall.c
* sysproc.c

فایل اضافه شده:

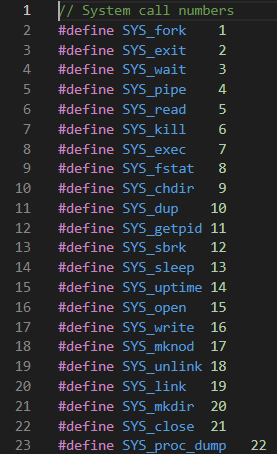
* ps.c

کد های موجود در هر فایل اضافه شده یا تغییر داده شده:

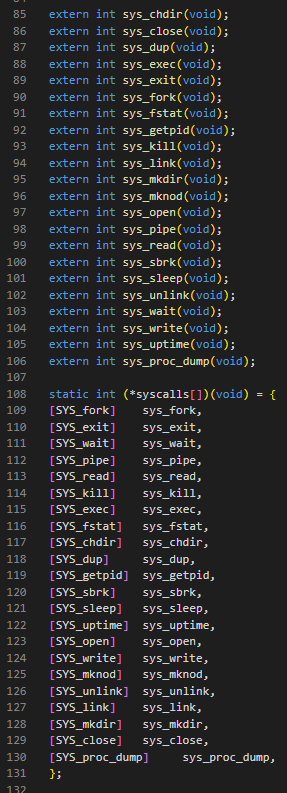
* defs.h



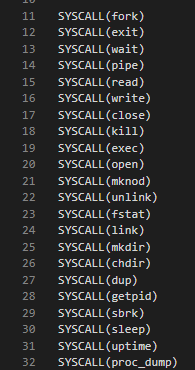
* syscall.h



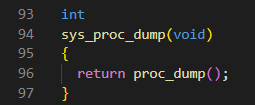
* syscall.c



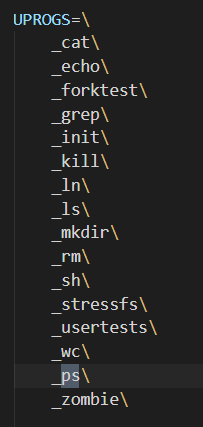
* usys.s

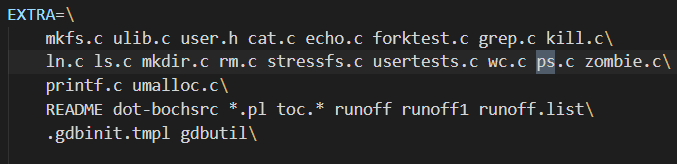


* sysproc.c



* Makefile

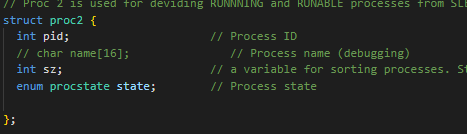




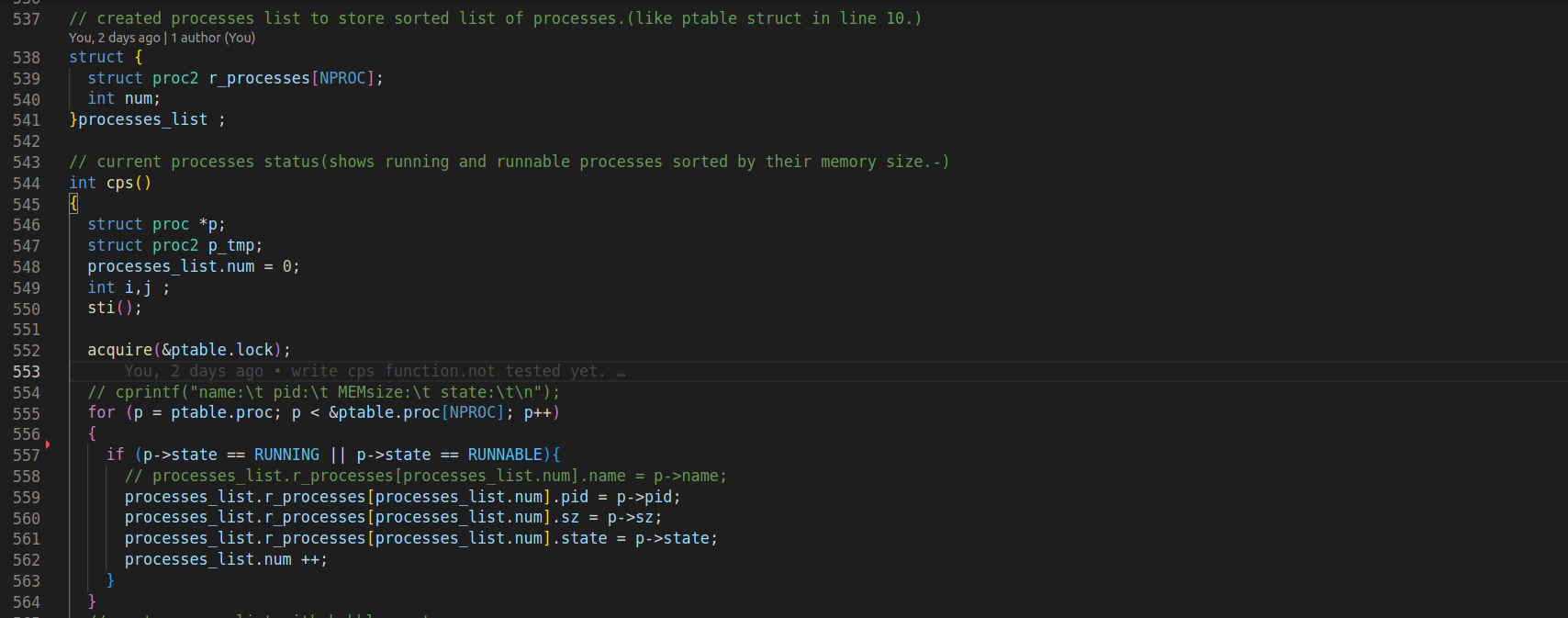
* ps.c

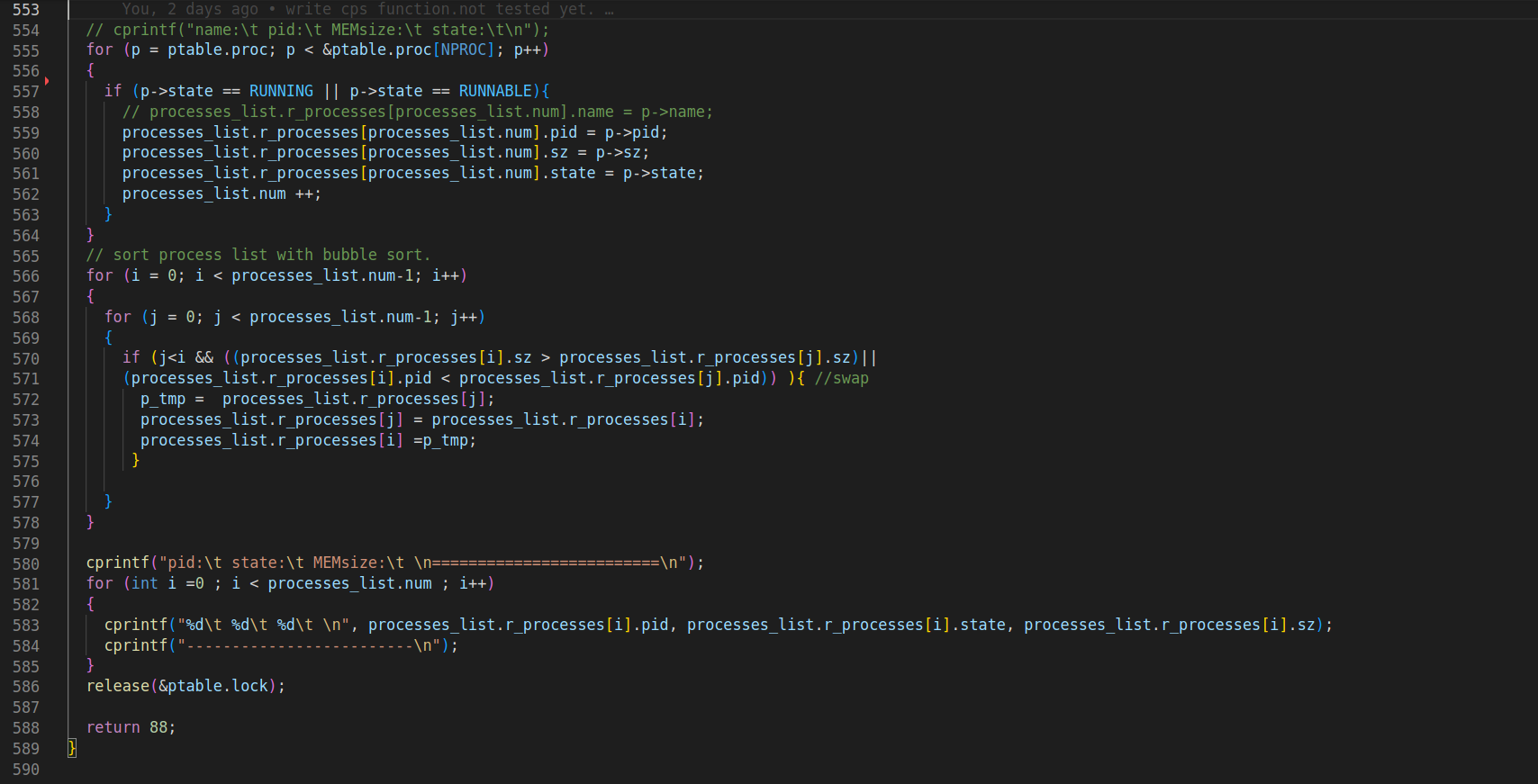


* proc.h



* proc.c





توضیح کلی از پروژه و کد ها و خروجی برنامه:

ابتدا سیستم کال درست میکنیم و کد های لازم برای ساختن سیستم کال را در فایل های مورد نیازش قرار میدیم.(همانطور که در ویدیوی توضیحات پروژه گفته شد.)

در فایل proc.h استراکتproc2 برای ذخیره اطلاعات پروسس ها را تعریف میکنیم که پروسس هایrunning ,runnable را با در آن ذخیره شود،مثل proc که خودش تعریف شده بود ، در آن pid,status,memory size پروسس ها ذخیره شوند.

در فایل proc.c استراکت دیگری میسازیم به اسمprocesses\_list که 2 ممبر آن برای ذخیره ی پروسس های خواسته شده و تعداد آنها است. در ادامه ی کار پروسس های running,runnable را جدا کرده،و در r\_proceesses داخل processes\_llist وارد میکنیم و تعداد آنرا هم 1 واحد زیاد میکنیم، و در لوپ بعدی با استفاده از bubble sort آن ها را سورت میکنیم. سپس هم در انتها همه ی آنهارا پرینت میکنیم.

در فایل ps.c یا همان فایل تست 10 فورک میزنیم(میتوان تعداد پروسس ها را به مقدار دلخواه تغییر داد) ویک عدد malloc برای تست کردن درست کار کردن سورت در سیستم کال.while(1) برای جلوگیری از این است که پروسس های زامبی تولید شوند.

منابع مهم:

[پروژه ی مشابه (از یوتیوب)](https://www.youtube.com/watch?v=21SVYiKhcwM)

[سیستم کال(GFG)](https://www.geeksforgeeks.org/introduction-of-system-call/)

[ریپازیتوری پروژه(گیتهاب)](https://github.com/mit-pdos/xv6-public)

* خروجی برنامه(با درستور ps ، و با 10 عدد فورک و1 malloc):

