

بسم الله الرحمن الرحيم

محمدرضا بابایی یزدلی – 97521126

فاطمه زهرا بخشنده – 98522157

• فایل های تغییر یافته :

.I	Syscall.c
.II	Syscall.h
.III	User.h
.IV	Defs.h
.V	Sysproc.c
.VI	Proc.c
.VII	Proc.h
.VIII	Makefile
.IX	Usys.s
.X	Test.c

در ابتدا شماره system call خود را در فایل Syscall.h تعریف میکنیم ، با وجود 21 دستور شماره این فراخوانی برابر 22 خواهد بود.

سپس باید اشاره گری برای فراخوانی موردنظر را در فایل Syscall.c که شامل آرایه ای از اشاره گر ها به توابع می باشد را اضافه کنیم.

در فایل Sysproc.c ما به پیاده سازی System Call خود میپردازیم. (sys_proc_dump)

در فایل Usys.s ما نام System Call ای که توسط کاربر صدا زده میشود را تعریف میکنیم.

در فایل های User.h و Defs.h ما به پیاده سازی اولیه و Struct موردنظر میپردازیم ، که آرگومان های این تابع * Struct proc_info و * int میباشد.

در اینجا چون میخواهیم از سطح kernel پاس بدهیم مانند user نمیشود پس ما در اینجا تابع `orgptr` که این پاس دادن ها در فایل `Sysproc.c` صورت میگیرد و در آن فایل از آن استفاده میکنیم.

در قسمت بعدی که بخشی مهم است به فایل `Proc.c` مراجعه میکنیم و شروط لازم

(`Running || Runnable`) را ایفا میکنیم و آرایه پوینتری را پر میکنیم و `size` پر شده را برمیگردانیم.

بعد از آن در `Sysproc.c` بر اساس `memsize` مرتب میشود.

نکته ای که باید اضافه کرد آن است که در فایلی به نام `Proc.h` اطلاعات خوبی به ما درباره `Struct` به نام `proc` میدهد که `size` همان `sz` و `pid` همان `pid` میباشد و همچنین ما در اینجا

```
struct proc_info {  
    int pid;  
    int memsize;  
}
```

را تعریف میکنیم.

در بعد از آن که شرط ها را در `Proc.c` پیاده سازی کردیم لازم است که `User Program` به نام `Test.c` بسازیم که در آن برای نمایش از `fork` و `malloc` استفاده کردیم و اندازه هرکدام را متفاوت قرار داده ایم بعد از آن `System Call` را فراخوانی میکنیم و یک `* struct proc_info` و آدرس یک `int` را پاس میدهیم.

بعد از پرکردن در ادامه جدول `sort` شده (مرتب سازی حبابی) را چاپ میکند.

برای استفاده و اجرا گرفتن باید چند متغیر در `Makefile` ایجاد کنیم.

نقاط تغییر یافته در `uprogs` که اسم فایل `user` را که `test` نام داشت اضافه شده و در قسمت `Extra` ، `Test.c` در کنار بقیه آمده و تغییرات را ثبت کرده و مشاهده میکنیم که به خوبی کار میکند.

• چالش ها :

پاس دادن اندازه `int` و `* struct` به تابع

مرتب سازی

تغییرات در فایل های ضروری جهت ایجاد شدن فراخوانی

- مرتب سازی به صورت نزولی و بر اساس اندازه میباشد و در صورت برابر بودن بر مبنای pid
- در Proc.c ، Lock داریم تا دو procces همزمان اجرا نشود.