فاز دوم پروژه سیستم عامل

در این فاز سعی بر پیاده سازی thread ها داشتیم. به طوری که عملا از پردازه ها به جای thread استفاده میکنیم. برای این کار به دو متغیر جدید در ساختار struct مربوط به process داریم. یکی برای تعداد thread ها و مشخص کردن اینکه آیا پردازنده current یک پردازه است یا thread. متغیر بعدی stackTop برای مشخص کردن stack هر thread است. به طوری که thread ها باید stack های مختلفی داشته باشند.

تغییرات خارج از ویدئو که داده شده در exec.c باید این دو متغیر مقدار دهی اولیه شوند. همچنین سیستم کال های مربوطه ساخته شده و روش مقدار دهی مناسبی برای برای توابع این سیستم کال ها نوشته شده.

همچنین در فایل proc.c هم در توابعی مانند wait و userinit و allocproc تغییراتی داده شده است. به طوری که دو متغیر جدید تعریف شده مقدار دهی اولیه شده.

در تابع exit باید چک کنیم که آیا curproc یک process است یا یک thread و اگر thread بود فقط خود آن thread را free کنیم و نه کل pagedir را، وگرنه تمام thread ها از دست میروند.

در تابع growproc بر حسب اینکه curproc برای فرزند است یا پدر تغییراتی دادیم تا مقدار مموری اضافه شده به درستی اعمال شود.

در تابع fork باید مقدار thread را 1 بگذاریم که نشان دهنده یک process است. نیازی به مقدار دهی کردن به stackTop نداریم زیرا در هنگام عمل fork هر تو thread از یک stackTop استفاده اولیه میکنند (مقدار stackTop یکسان دارند)

در تابع wait هم باید چک کنیم اگر curproc معادل پردازنده پدر بود و پردازنده های فرزندی به عنوان thread داشت آن را free نکنیم وگرنه اطلاعات آن از دست میرود.

تابع جدید تعریف شده clone همانند thread_creator دستور کار عمل میکند به طوری که به جدا سازی stackTop ها دو thread مجزا میسازد که به والد assign میکند. این تابع مانند تابع fork عمل میکند با کمی تغییراتی که داده شده است.

تابع join هم مانند wait عمل میکند با تغییرات جزئی که قابلیت پشتیبانی از thread را به ما میدهد.