

گزارش پروژه اول OS

محمد اصولیان ۹۹۵۲۱۰۷۳ – محمدحسین عباسپور ۹۹۵۲۱۴۳۳

مقدمه

در این پروژه قصد داشتیم تا در سیستم عامل آموزشی xv6 که توسط MIT ساخته شده، یک syscall ایجاد کنیم که در آن پروسس های در حال اجرا در سیستم عامل گرفته شده و بر حسب حجم اشغالیشان مرتب می شوند و در ترمینال نمایش داده می شوند.

مراحل انجام پروژه

۱. نصب xv6 و پکیج های لازم
۲. افزودن عنوان syscall و تعیین id برای آن
۳. اضافه کردن دستور اجرای syscall مورد نظر به makefile
۴. اختصاص function برای عنوان تعریف شده
۵. تعریف توابع و ساخت تابع کمکی برای دریافت لیست پروسس ها و مرتب کردن آنها
۶. نوشتن تابع تست

۱- نصب xv6 و پکیج های لازم

برای نصب xv6 لازم بود تا از محیط لینوکس برای اجرای دستورات آن استفاده کنیم. بنابراین از wsl در ویندوز استفاده شد.

سیستم عامل و پکیج های آن با کمک دستورات موجود در مستند پروژه نصب شدند.

۲- افزودن عنوان syscall

برای عنوان syscall, یک id در فایل syscall.h در نظر میگیریم.

```
19 #define SYS_unlink 18
20 #define SYS_link 19
21 #define SYS_mkdir 20
22 #define SYS_close 21
23 #define SYS_proc_dump 22
24
```

۳- افزودن دستور اجرای syscall به فایل makefile

عنوان syscall را به آرایه UPROGS در فایل makefile اضافه میکنیم.

```
_usertests\
_wc\
_zombie\
_proc_dump\
```

۴- اختصاص فانکشن برای syscall

مشابه سایر systemcall ها نام این سیستم کال هم اضافه میکنیم.

```
extern int sys_unlink(void);
extern int sys_wait(void);
extern int sys_write(void);
extern int sys_uptime(void);
extern int sys_proc_dump(void);
```

```
[SYS_unlink] sys_unlink,
[SYS_link] sys_link,
[SYS_mkdir] sys_mkdir,
[SYS_close] sys_close,
[SYS_proc_dump] sys_proc_dump,
};
```

```
93 int
94 sys_proc_dump(void)
95 {
96     return proc_dump();
97 }
```

۵- تعریف توابع

ابتدا یک Struct به نام Proc_info برای ذخیره سائز و id پروسس میسازیم. سپس یک آرایه برای ذخیره تمام پراسس های RUNABLE و RUNNING ایجاد میکنیم.

در تابع proc_dump در فایل proc.c با استفاده از ptable تمام پراسس ها را دریافت کرده و پراسس های RUNABLE و RUNNING را جدا میکنیم و در آرایه process میریزیم. سپس در تابع sort_process این پراسس ها را بر اساس size و سپس id مرتب کرده و نمایش میدهیم.

۶- نوشتن تابع تست

برای تست توابع نوشته شده، تابع main در فایل proc_dump پیاده سازی شده. برای اجرای تست کاربر باید پس از دستور make qemu یا make qemu-nox، دستور

proc_dump NUMBER_OF_PROCESSES SIZE_OF_PROCESS

را اجرا کند. (مقادیر پیشفرض $NUMBER_OF_PROCESSES=10$ و $SIZE_OF_PROCESS=5000$ هستند).

با انجام این دستور یک حلقه $NUMBER_OF_PROCESSES$ بار اجرا میشود و در هر بار یک فورک زده شده یک یک پروسس در آن اجرا میشود. پس از اجرای تمام حلقه ها، در آخر `main` سیستم کال نوشته شده صدا زده میشود و نتایج در ترمینال نمایش داده میشوند.