سیستم‌های‌عامل

بهار 1403

دکتر نوروززاده

تمرین شماره دو

1. اقدامات انجام شده توسط یک هسته برای Switching Context بین فرآیندها را توضیح دهید. چه تفاوتی با اقدامات انجام شده برای context switching بین نخ‌های سطح هسته (KLT) وجود دارد؟

2. معنای "دقیقا یک بار" یا once exactly را با توجه به مکانیسم RPC در نظر بگیرید. آیا حتی اگر پیام ACK ارسال شده به کلاینت به دلیل مشکل شبکه از بین برود الگوریتم اجرایی این مفهوم به درستی اجرا می شود؟ توالی پیام ها را توصیف کنید و در مورد اینکه آیا "دقیقا یک بار "هنوز حفظ شده است یا خیر توضیح دهید.

3. نوعی از زمانبندیRR ، زمانبندی RR رگرسیون (robin-round regressive) است. این زمانبندی به هر فرآیند یک کوانتوم زمانی و یک اولویت اختصاص می‌دهد. مقدار اولیه یک کوانتوم زمانی 55 میلی‌ثانیه است. با این‌حال، هر بار که یک فرآیند به CPU اختصاص داده می‌شود و از کل کوانتوم زمان خود استفاده می‌کند (برای I/O مسدود نشود)، ۱۵ میلی‌ثانیه به کوانتوم زمان و اولویت آن اضافه می‌شود(کوانتوم زمانی براي یک فرآیند را می‌توان حداکثر تا ۱۵۵ میلی‌ثانیه افزایش داد). اگر که یک فرآیند قبل از استفاده از کل کوانتوم زمانی خود مسدود شود، کوانتوم زمانی آن 5 میلی‌ثانیه کاهش می‌یابد، اما اولویت آن ثابت می‌ماند. این نوع زمانبندی به نفع چه نوع فرآیندی (CPU Bound or I/O Bound) است؟ توضیح دهید.

4. در یک سیستم بالدرنگ سخت، حداکثر چند فرایند با زمان اجرای 20ms و دوره تناوب 100ms قابل زمان بندی هستند بطوری که زمان switching Context را 5ms در نظر بگیریم.

5. تیکه کد زیر را در نظر بگیرید:

```
pid_t pid;

pid = fork();
if (pid == 0) { //Child process
fork();
thread_create(...);
}
fork();
```

- چه تعداد فرآیند ساخته می‌شود؟
- چند ترد ساخته می‌شود؟

6. با استفاده از برنامه شکل ،4 مقادیر pid را در خطوط A ، B، Cو D شناسایی کنید.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
int main()
{
    pid t pid, pid1;
    /* fork a child process */
    pid = fork();
    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        pid1 = getpid();
        printf("child: pid = %d",pid); /* A */
        printf("child: pid1 = %d",pid1); /* B */
    }
    else { /* parent process */
        pid1 = getpid();
        printf("parent: pid = %d",pid); /* C */
        printf("parent: pid1 = %d",pid1); /* D */
        wait(NULL);
    }
    return 0;
}
```

7. با استفاده از قانون Amdahl ، مقدار افزایش سرعت را برای هرکدام از شرایط زیر محاسبه کنید:
   - 40 درصد موازی با (a) 8 هسته پردازشی و (b) 16 هسته پردازشی.
   - 67 درصد موازی با (a) 2 هسته پردازشی و (b) 4 هسته پردازشی.
   - 90 درصد موازی با (a) 4 هسته پردازشی و (b) 8 هسته پردازشی.

8. چه بخش‌هایی از program state در بین تردهای مختلف یک برنامه مولتی‌تردی مشترک‌اند؟

9. توضیح دهید چرا سوییچ کردن بین تردهای یک برنامه از سوییچ کردن بین دو فرآیند سریع تر است؟

10. تحت چه شرایطی یک راه حل چند نخی با استفاده از نخ های هسته چندگانه عملکرد بهتری نسبت به راه حل تک نخی در یک سیستم تک پردازنده ای ارائه می دهد؟

موفق باشید