مهدی محمدی نسب ۹۶۳۱۱۲۱۰۱

۱. ۱. روش interrupt-based I/O: وقتی مشکلی در سخت افزار های ورودی و خروجی ایجاد شده اطلاعی به پردازنده داده می شودتا آن را بررسی کد و اقدام لازم را انجام دهد.

۲ . polling- based I/O(مبتنی بر سرکشی): دربازه های خاصی وضعیت هر قطعه بررسی میشود تا ازرخداد خطا مطلع شود.

۲. روش debugging:روش coredump: مربوط به پردازه است. فایلی است که فضای آدرس پردازه(حافظه) و مقدار رجیستر هارا درهنگام توقف غیرمنتظره پردازه در خود ذخیره میکند. ممکن است بعد از خاتمه یافتن پردازه تولید شوند. توسط kernel در پاسخ به کرش های برنامه راه اندازی می شوند.

روش crash dump: مربوط به هسته است. و مانند coredump حافظه و مقدار رجیسترها در این فایل ذخیره میشوند.

Kernel panic: اصطلاح کرنل دستپاچه شده. درصورت بروز مشکل برای بدتر نشدن وضعیت کاری انجام نمیدهد.

روش profiler: حل و یافتن مشکلات performance. بهتر و سریعتر کرد گلوگاه. برای بهترکردن استفاده از حلفظه یا cache یا سرعت یا خطای صفحه.

روش logging: گزارش رخداد های یک سیستم

۳. Shared memory: از حافظه مشترک استفاده میکنند. پیاده سازی آن نیازی به استفاده از کتابخانه و توابع خاصی ندارد.نسبت به مدل دیگر سریعتر است. امکان خطا در پیاده سازی آن بیشتر است و در کد ممکن است مشکلاتی به وجود آید.

Message passing: از کتابخانه هایی باید برای این مدل استفاده کرد. توابع send و cons استفاده میشوند. برنامه آن خواناتر و ساده تر است و احتمال بروز خطا کمتر است.از یک کلید یا شناسه برای انتقال اطلاعات و دریافت استفاده میکند.

۴. هدف استفاده ایجاد پردازه جدید است. با استفاده از تابع ()exec در برنامه ها به کار می رود. برای تعامل بین چند پردازه مثل هم و اجرای یک شبه کد برای چند بار واجرای موازی و همزمان چند پردازه و همچنین اجرای یک پردازه دیگر در قسمتی از کد با اجرا ادامه کد ها که با استفاده از exec انجام میشود.

۵. اگر یک پردازه پدر چند پردازه فرزند داشته باشد، اگر پردازه ای خاتمه یابد ولی پدرش هنوز waitرا برای آن فراخوانی نکرده باشد، اصطلاح zombie را برای آن فرزند به کار میبرند.با صدا زده شدن wait توسط parent آن zombieاز بین میرود، درصورتی که اصلا wait فراخوانی نشود init فرزند zombie شده را قبول کرده و سپس به طور خودکار برای ناپدید شدن wait را صدا میزند

۶. خیر، درمواردی اطلاعات آن به این دلیل که ممکن است مورد استفاده قرارگیرد ازبین نمیرود مثلا در حالت زامبی که منابع فرزند از بین رفته برای بقیه پردازه ها آزاد میشوند. حالتی که پردازه والد زودتر میمیرد را orphan گویند.

۷. Init باشناسه ۱ که توسط سیستم عامل در هنگام بوت شدن سیستم به وجود می آید و تاوقتی که خاموش نشود به عملکرد خود ادامه میدهد.درصورت مرگ init سیستم خاموش میشود.

۸. درابتدا پدر با i=0است. سپس در اولین اجرا یک فرزندبا i=0 میسازد. سپس در تکرار بعدی iهردو یک شده و هردو یک فرزند میسازند که ۴ پردازه تا کنون داریم. درتکرار بعدی برای هر ۴ پردازهi=2 میشود و سپس هر ۴ پردازه یک پردازه تولید کرده و همینطور تا i=4 .که در نهایت ۱۶ پردازه داریم.

۹. ۵ چاپ میشود.

10.

```
int main()
{
    int counter = 0;
    if(fork()>0)
    {
        counter++;
        cout<<"A/"<<counter<<endl;
        if(fork() == 0)
        {
            counter++;
            cout<<"C/"<<counter<<endl;
            if(fork()>0)
                wait(NULL);
            else
                cout<<"D/"<<counter<<endl;
        }
        else wait(NULL);
    }
     else
    {
        cout<<"B/"<<counter<<endl;
        if(fork() == 0)
        {
            counter++;
            cout<<"C/"<<counter<<endl;
            if(fork()>0)
                wait(NULL);
            else
                cout<<"D/"<<counter<<endl;
        }
        else wait(NULL);
    }
```