

**تکلیف اول درس سیستم عامل**  
**پاییز ۱۴۰۰ - گروه دکتر زینب زالی**  
**دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان**

---

۱- هر یک از دسته مفاهیم زیر را تعریف کرده و باهم مقایسه کنید:

الف) multiprocessor system و clustered systems

ب) system call و system call API

ج) برنامه بوت لودر (مانند Grub) و برنامه بوت سیستم (مانند UEFI)

۲- با توضیح مختصر و کافی مشخص کنید هر یک از موارد زیر درست یا نادرست است؟

الف) روش shared memory در IPC از دید کاربر برای برنامه‌نویسی سخت تر از message passing است

ب) PCB یک ساختار داده از اطلاعات مفید و مهم هر پروسس است که در فضای آدرس آن پروسس نگهداری می‌شود

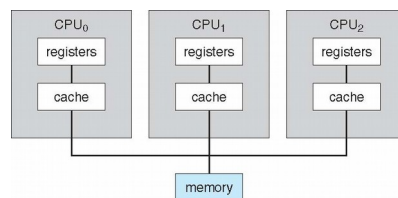
۳- مجازی سازی (Virtualization) در محاسبات ابری (Cloud Computing) چه نقشی دارد؟

می‌توان گفت مدیریت نرم‌افزاری cloud مانند یک سیستم عامل است. دو وظیفه اصلی سیستم عامل ایجاد تجرید در استفاده از سخت افزار و resource management (مدیریت منابع) است. نقش و اهمیت این دو وظیفه را در cloud بیان نموده و آن را با سیستم عامل یک کامپیوتر شخصی مقایسه کنید.

۴- آیا عملیات همزمان اجرای برنامه توسط CPU و انتقال داده توسط DMA ممکن است تداخلی باهم داشته باشند؟ در صورت مثبت بودن جواب، این اتفاق در چه شرایطی رخ می‌دهد؟ بین CPU و DMA اولویت دسترسی به حافظه اصلی با کدام یک است؟ چرا؟

۵- ساختارهای یکپارچه (monotonic) و ماژولار را در از نظر کارایی (performance) و انعطاف‌پذیری مقایسه کنید.

۶- در یک سیستم SMP مطابق شکل زیر، با یک مثال توضیح دهید چگونه ممکن است داده موجود در حافظه، مقادیر مختلفی در کشهای محلی داشته باشد.



۷- الف) با رسم شکل نشان دهید یک فرآیند در حافظه اصلی کامپیوتر شامل چه بخش‌هایی می‌باشد

ب) تعویض متن به چه منظور انجام میشود؟ PCB در این عملیات چه نقشی دارد؟

ج) برای تغییر وضعیت یک پروسس از حالت اجرا به انتظار دو عامل که یکی به اختیار پروسس و دیگری بدون اختیار او رخ می‌دهد نام ببرید.

## ۸- برنامه روبرو را در نظر بگیرید

```
int a = 5;
int fd = open(...) //opening a file
int ret = fork();
if(ret > 0) {
    close(fd);
    a = 6;
    ...
}
else if(ret==0) {
    printf("a=%d\n", a);
    read(fd, something);
}
```

بعد از اینکه پروسس جدید fork شد، فرض کنید ابتدا پروسس والد قبل از پروسس فرزند زمان‌بندی می‌شود (نوبت اجرا در cpu می‌گیرد). همچنین پروسس فرزند برای اولین بار بعد از اجرای کامل پروسس والد زمان‌بندی می‌شود. به سؤالات زیر با توضیح مختصر پاسخ دهید:

الف) مقدار a که در پروسس فرزند چاپ می‌شود چند است؟

ب) آیا تلاش برای خواندن از fd در پروسس فرزند موفقیت‌آمیز خواهد بود؟

ج) با ترتیب زمان‌بندی که بیان شد، آیا پروسس فرزند روح سرگردان می‌شود یا یتیم یا هردو؟ با اضافه کردن کد مناسب به این برنامه، از این اتفاق یا اتفاقات جلوگیری کنید.

۹- کد زیر قسمتی از برنامه shell برای اجرای پایپلاین دو دستور (مانند `grep new | ls` برای لیست کردن فایل‌هایی که نامشان شامل new است) است. با استفاده از دانش خود و جستجوی دستورات و `system call`‌های ناآشنا توضیح دهید خطوط مشخص شده چه می‌کنند و در کل توضیح دهید این کد چگونه اجرای پایپلاین دو دستور را پیاده‌سازی می‌کند.

```
void makePipeline(char* command1, char** argv1, char* command2, char** argv2){
    int fd[2];
    pipe(fd);
    pid_t pidComm[2];
    pidComm[1] = fork();
    if(pidComm[1] == -1){
        printf("Can't Create New process for pipeline\n");
        return;
    }else if(pidComm[1] == 0){
        close(fd[0]);
        dup2(fd[1], 2);
        dup2(fd[1], STDOUT_FILENO);
        execv(command1, argv1);
        printf("%s: No such command\n", command1);
        exit(0);
    }else if(pidComm[1] > 0){
        pidComm[0] = fork();

        if(pidComm[0] < 0){
            printf("Can't Create New Process for pipeline\n");
            kill(pidComm[1], SIGKILL);
            return;
        }else if(pidComm[0] == 0){
            close(fd[1]);
            dup2(fd[0], 0);
            dup2(fd[0], STDIN_FILENO);
            execv(command2, argv2);
            printf("%s: No such command\n", command2);
            exit(0);
        }else if(pidComm[2] > 0){
            //parent Never use the pipe:
            close(fd[0]);
            close(fd[1]);
            wait(NULL);
            wait(NULL);
        }
    }
    return;
}
```