

1. الف) سیستم های چند پردازنده ای: یک سیستم که شامل 2 یا تعداد بیشتری processor است. سیستم های خوشه ای، متشکل از چند سیستم که معمولاً به یک شبکه ذخیره سازی مشترک متصل هستند.

ب) سیستم کال متودی است که با استفاده از آن، یک پروسه به صورت امن و کنترل شده به سیستم عامل درخواست انجام کاری میدهد، مانند عمل باز کردن فایل.

رابط برنامه نویسی سیستم کال، راهی است که سیستم عامل در اختیار کاربر برای استفاده از سیستم کال ها قرار میدهد که کاربر اطلاعی از چگونگی پیاده سازی آن ندارد و فقط نحوه استفاده از آن را میداند، مانند تابع open.

ج) بوت سیستم اولین برنامه ای است که پس از روشن شدن اجرا میشود و در ROM قرار دارد. بوت لودر برنامه ای است که قبل از کرنل اجرا میشود و رابط بین بوت سیستم و سیستم عامل است.

2. الف) درست است، زیرا در این روش برنامه نویس باید سینک بودن عملیات هارا کنترل کند و جلوی Race Condition را بگیرد

ب) غلط است، PCB در داخل کرنل نگه داری میشود

3. محاسبات ابری، قدرت محاسباتی، فضای ذخیره سازی و حتی برنامه های کاربردی را به عنوان یک سرویس ارائه میدهد، مجازی سازی امکان مدیریت و ارائه این منابع را در سراسر شبکه فراهم میکند.

به این صورت که از مجموعه زیادی از سرور ها، ماشین ها و فضاهای ذخیره سازی، کاربر فقط دسترسی به بخشی محدود و مجازی را دارد، مانند سیستم عامل که به ظاهر هر نرم افزار بر روی یک کامپیوتر با پردازنده و حافظه اختصاصی خودش اجرا میشود.

4. بله، اگر هر دو نیاز به دسترسی به یک بلاک حافظه داشته باشند، یکی برای پردازش و دیگری برای انتقال، تداخل ایجاد میشود.

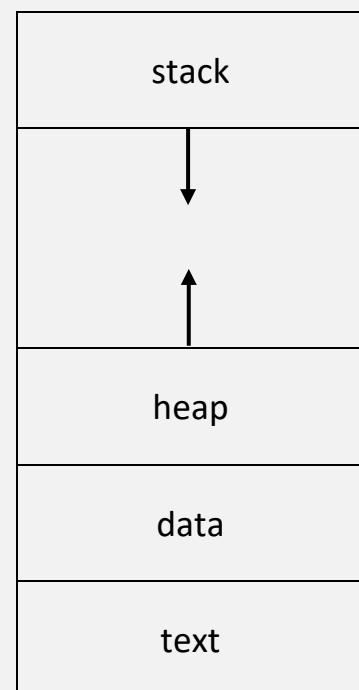
اولویت با DMA است، زیرا در غیر این صورت overrun یا underrun رخ میدهد و ورودی و خروجی دستگاه ها با مشکل مواجه میشود، به طور مثال ورودی به موقع از بافر دستگاه به سیستم نمیرسد و از بین میرود.

5. ساختار ماژولار انعطاف پذیری بیشتری دارد، زیرا برای تغییر یا اضافه کردن یک بخش به کرنل، نیاز به تغییر کل کرنل نیست و فقط کافیسست ماژولی تغییر کند یا اضافه شود.

ساختار یکپارچه از نظر کارایی سریعتر است زیرا تمامی رابط ها و سرویس ها در یک فضای آدرس پیاده سازی شده و اجرا میشوند.

6. زمانی که یک برنامه multithread باشد و یک متغیر در چند ترد همزمان استفاده شود، احتمال این اتفاق وجود دارد

7. الف)



ب) به منظور scheduling پروسه ها انجام میشود، PCB برای ذخیره وضعیت پروسه در حال اجرا و بارگذاری وضعیت پروسه بعدی برای اجرا استفاده میشود

ج) به اختیار: سیستم کال sleep، بدون اختیار: تمام شدن مهلت اجرا (به دست scheduler)

8. الف) مقدار 5، حافظه a بین والد و فرزند مشترک نیست، برای همین تغییر در والد تاثیری در متغیر فرزند ندارد

ب) بله

ج) هم زامبی میشود و هم یتیم، زیرا پروسه والد بدون اینکه فرزند را reap کند (تابع wait فراخوانی نشده) تمام میشود

```
int a = 5;
int fd = open(...)
int ret = fork();
if(ret > 0){
    close(fd);
    a=6;
    wait(0);
}
else if(ret == 0){
    printf("%d",a);
    read(fd,something);
    exit(EXIT_SUCCESS);
}
```

9. سیستم کال pipe، یک ordinary pipe باز میکند که fd[0] سمت خواندن و fd[1] سمت نوشتن آن است.

سیستم کال dup2 این اجازه را میدهد که یک کپی فایل دیگر باشد، به عبارت دیگر پس از صدا زدن این تابع بر روی دو file descriptor، هر تغییری بر روی یکی از آن ها بر دیگری نیز اعمال میشود در اینجا این تابع بر روی fd[1] و STDOUT_FILENO صدا زده شده، به این معنی است که تمامی خروجی های این پروسه بر روی پایپ نیز نوشته میشود

پس از آن برنامه مورد نظر و آرگومان های آن با استفاده از execv اجرا میشود و در حافظه فرزند قرار میگیرد، همینطور تمامی خروجی های آن در پایپ ریخته خواهد شد

برای پروسه فرزند دوم fd[0] و STDIN_FILENO کپی شده اند، به این معنی است که تمامی خروجی های پروسه فرزند اول، به عنوان ورودی به پروسه فرزند دوم داده میشود، پس از آن برنامه دوم و آرگومان های آن با دستور execv اجرا میشود و ورودی خود را از پایپ میگیرد

در نهایت والد با صدا زدن سیستم کال wait، منتظر تمام شدن پروسه فرزندان بوده و حافظه آن ها را آزاد میکند