به نام خدا

امتحان پایان ترم سیستم های عامل  مصطفی فضلی شهری – 9822803  1400/4/5

**-1**

**با توجه به تعریف فراخوانی سیستمی، در مواردی که بین user mode وkernel mode سوئیچ می شویم، فراخوانی سیستمی را انجام می دهیم؛ معمولا سیستم عامل ها یک کتابخانه توابع دارند که درون آن فراخوانی های سیستمی صدا زده می شود مانند کتابخانه استاندارد C که شامل دستورات مختلفی اعم از printf و scanf و ... است.**

**در خط های :**

**4 عبارت printf فراخوانی سیستمی مدیریت و کار با خروجی، برای چاپ عبارت در خروجی استفاده می شود.**

**5 عبارت scanf فراخوانی سیستمی مدیریت و کار با ورودی، برای گرفتن ورودی از کاربر استفاده می شود.**

**6 عبارت fopen برای فراخوانی سیستمی مدیریت فایل و به صورت دقیق تر خواندن از فایل استفاده می شود.**

**8 عبارت printf فراخوانی سیستمی مدیریت و کار با خروجی، برای چاپ عبارت در خروجی استفاده می شود.**

**9 عبارت exit فراخوانی کنترل فرآیند و دقیقتر بستن فرآیند با آرگومان 0 صدا زده می شود.**

**11 عبارت printf فراخوانی سیستمی مدیریت و کار با خروجی، برای چاپ عبارت در خروجی استفاده می شود.**

**12 عبارت scanf فراخوانی سیستمی مدیریت و کار با ورودی، برای گرفتن ورودی از کاربر استفاده می شود.**

**13 عبارت fopen برای فراخوانی سیستمی مدیریت فایل و به صورت دقیق تر خواندن از فایل استفاده می شود.**

**15 عبارت printf فراخوانی سیستمی مدیریت و کار با خروجی، برای چاپ عبارت در خروجی استفاده می شود.**

**16 عبارت exit فراخوانی کنترل فرآیند و دقیقتر بستن فرآیند با آرگومان 0 صدا زده می شود.**

**18 عبارت fgetc برای کار با ورودی ، فراخوانی سیستمی مربوطه را برای گرفتن صدا می زند.**

**20 عبارت fputc برای کار با خروجی فراخوانی سیستمی مربوطه را برای قرار دادن عبارت صدا میزند.**

**23 عبارت printf فراخوانی سیستمی مدیریت و کار با خروجی، برای چاپ عبارت در خروجی استفاده می شود.**

**24 عبارت fclose برای بستن فایل، فراخوانی سیستمی مربوطه را صدا می زند.**

**25 عبارت fclose برای بستن فایل، فراخوانی سیستمی مربوطه را صدا می زند.**

**– 2**

**a = 10**

**b = 2**

**برای این دو فرآیند که هر کدام دو خط دارند، چهار حالت داریم:**

**1- در ابتدا فرآیند P1 و سپس فرآیند P2 به صورت کامل اجرا شوند که بدین ترتیب داریم :**

**P1 :**

**b = b + 2 => b = 4**

**a = a – b => a = 6**

**P2:**

**b = b – 1 => b = 3**

**a = a + b => a = 9**

**که در آخر a = 9 و b = 3 را داریم.**

**2- ابتدا فرآیند P2 به صورت کامل و سپس فرآیند P1 به صورت کامل انجام شوند.**

**P2 :**

**b = b – 1 => b = 1**

**a = a + b => a = 11**

**P1:**

**b = b + 2 => b = 3**

**a = a – b => a = 8**

**که مقادیر a و b در انتها به ترتیب برابر 8 و 3 هستند.**

به نام خدا

امتحان پایان ترم سیستم های عامل  مصطفی فضلی شهری – 9822803  1400/4/5

**3- به صورت خط به خط ابتدا خطی از P1 اجرا شود و سپس خطی از P2 اجرا شود که داریم :**

**P1:**

**b = b + 2  => b = 4**

**P2:**

**b = b – 1  => b = 3**

**P1:**

**a = a – b  => a = 7**

**P2:**

**a = a + b  => a = 10**

**که در انتها مقادیر a و b به ترتیب برابر 10 و 3 می باشند.**

**4- به صورت خط به خط ابتدا خطی از P2 اجرا شده و سپس خطی از P1 اجرا شود که داریم :**

**P1:**

**b = b – 1  => b = 1**

**P2:**

**b = b + 2  => b = 3**

**P1:**

**a = a + b  => a = 13**

**P2:**

**a = a – b  => a = 10**

**که در این روش همانند روش قبل در آخر a و b به ترتیب برابر 10 و 3 هستند.**

**در کل b میتواند عدد 3 و a می تواند عدد های 8، 9 ، 10 باشند که این اعداد به طریقه اجرا شدن و ترتیب اجرا شدن کد بستگی دارد.**

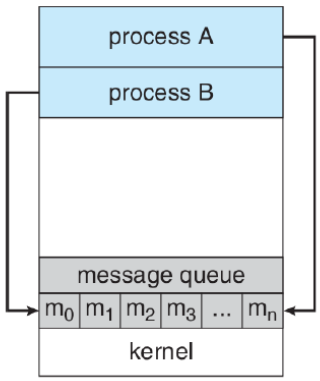
**3 -**

**با توجه به مشابهت و استفاده mutexLock از دستورات test_and_set استفاده می کنیم، پس این دو در ویژگی انتظار محدود نیز باید مشابهت داشته باشند، همانطور که در کلاس گفته شد در سیستم های دارای پردازنده تک هسته ای هیچ تضمینی وجود ندارد که فرآیندی وارد ناحیه بحرانی شود و پس از خارج شدن از ناحیه بحرانی، دوباره خود وارد ناحیه بحرانی نشود؛ بدین مفهوم که یک فرآیند بارها و بارها و به صورت پیاپی پشت سر هم اجرا شود بدون آنکه نوبت و اولویت در ترتیب اجرای فرآیند ها رعایت شود.**

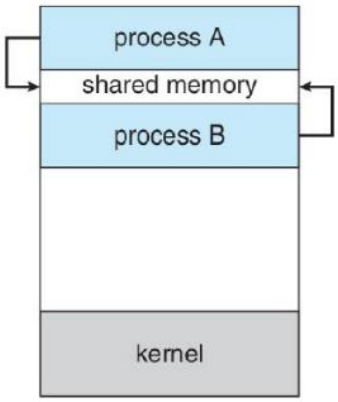
- 4

**برای اشتراک گذاری داده بین فرآیند ها دو راه حافظه مشترک و ارسال پیام داریم:**

**حافظه مشترک : بخشی از حافظه اصلی یا رم، بین فرآیند های همکار با استفاده از فراخوانی سیستمی به اشتراک گذاشته می شود. در این حالت هر دو فرآیند خانه ای از حافظه را به صورت مشترک قرار می دهند و با استفاده از فراخوانی سیستمی از آن میخوانند یا بر روی آن می نویسند و اینگونه آن خانه حافظه بین هر دو فرآیند به اشتراک گذاشته می شود.**




**ارسال پیام: در این روش اشتراک گذاری داده از طریق ارسال پیام انجام می شود و هر ارسال و دریافت پیام نیاز به فراخوانی سیستمی دارد، فرستنده در این حالت، پیام ها را در یک صف به هسته سیستم عامل ارسال می کند و خواننده پیام ها را از هسته سیستم عامل می خواند. در گذشته به دلیل سربار بالا از این روش کمتر استفاده می شد اما امروز به دلیل وجود پردازنده های چندهسته ای دارای Cache های بالا، از این روش نیز استفاده می شود.**



به نام خدا

امتحان پایان ترم سیستم های عامل مصطفی فضلی شهری – 9822803 1400/4/5

در روش Shared Memory بخشی از حافظه اصلی بین فرآیند های همکار به اشتراک گذاشته می شود ولی در روش Message Passing اشتراک گذاری داده از طریق ارسال و دریافت پیام انجام می شود .

در روش Shared Memory هر دو فرآیند فقط به آدرس خود دسترسی دارند و هر آدرسی که درخواست بدهند، به حافظه خودشان بر میگردند ولی در روش Message Passing فرستنده پیام ها را در یک صف به هسته سیستم عامل ارسال می کند.

در هر دو روش نیاز به فراخوانی های سیستمی داریم؛ در روش Shared Memory فقط برای راه اندازی به فراخوانی سیستمی نیاز داریم و پس از آن هر دو فرآیند به حافظه مشترک بدون نیاز به فراخوانی سیستمی دسترسی دارند ولی در روش Message Passing برای هر ارسال و دریافت پیام نیاز به فراخوانی سیستمی داریم.

در روش Message Passing خواننده پیام ها را از هسته سیستم عامل می خواند و در گذشته به دلیل سربار بالا، کمتر از این روش استفاده می شد ولی امروزه با روی کار آمدن پردازنده های چند هسته دارای Cache های بالا، از این روش نسبت به قبل بیشتر استفاده می شود.