آزمونک ۱ درس سیستمعامل (کارشناسی)			
تاریخ: ۱۴۰۲/۰۸/۰۷ مدت: ۱ ساعت			
شماره دانشجویی:		نام و نام خانوادگی:	

۱- کدام دو خصیصه از مهمترینهای دلایل وجود سیستم عامل است؟

الف)

ب)

۲- کدام درست و کدام نادرست است؟

الارست؟ با المهيوترها بدون سيستمعامل هم ميتوانند كار كنند. اجرا دستورات بر روى پردازنده بر عهده سيستم عامل است. اجرا دستورات بر روى پردازنده بر عهده سيستم عامل است. كنترل و مديريت نحوه اجراى برنامههاى كاربر تماماً توسط سيستم عامل انجام مىشود. كامپيوترهايى كه DMA دارند، از سرعت بيشترى هنگام اجراى برنامهها برخوردارند. كامپيوترهايى كه DMA دارند، از سرعت بيشترى هنگام اجراى برنامهها برخوردارند. كامپيوترهاى انبوهى (Batch computers) همان كامپيوترهاى پردازش سنگين (High computing) همان كامپيوترهاى پردازش سنگين و در زمانى كه برنامههاى كاربر وجود ندارد، سيستم عامل است. در زمانى كه برنامههاى كاربر وجود ندارد، سيستم عامل بصورت دايم در حلقه بينهايت در حال چک كردن است، زيرا در زمانى كه برنامه كاربر نيست، اين سيستم عامل است كه داخل حلقه فون-نيومن در حال اجرا است. وجود وقفه در پردازندهها براى سيستم عامل اجبارى است. وجود وقفه در پردازندهها بسيستم عامل را از اجراى كارهاى كاربر و ديگر فعاليتها باخبر مىكند. وجود وقفه در پردازندهها سيستم عامل را از اجراى كارهاى كاربر و ديگر فعاليتها باخبر مىكند. امروز سيستمهاى عامل قديمى بيشتر Time-Driven هستند. سيستمهاى عامل امروزى بيشتر Time-Driven هستند. سيستمهاى عامل امروزى بيشتر Time-Driven هستند. سيستمهاى عامل بدون قابليتهاى خاص خود كه بايد توسط توليدكنندگان پردازنده ايجاد شود، قادر به كنترل تمام اتفاقات داخل كامپيوتر نيستند. دروشى وجود دارد كه كاربر قادر است كه برنامه خود را در حالت Actil انجام مىشود. تغيير حالت از Sald بر ساسو و Kernel mode على الاحبارى انجام مىشود.		
اهرست؟ المربوترها بدون سیستم عامل هم میتوانند کار کنند. اجرا دستورات بر روی پردازنده بر عهده سیستم عامل است. اجرا دستورات بر روی پردازنده بر عهده سیستم عامل است. کنترل و مدیریت نحوه اجرای برنامه طولانی نیست که مدام کارهای سیستم را مدیریت می کند. کامپیوترهایی که DMA دارند، از سرعت بیشتری هنگام اجرای برنامهها برخوردارند. مدیریت سیستم فایل و مدیریت کاربران از وظایف سیستم عامل است. کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing) همان کامپیوترهای پردازش سنگین (CPU Utilization) در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. است. اب در نظر گرفتن سهم سیستم عامل، همیشه میزان بهره گیری از CPU Utilization) CPU درصد است. زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل اجباری است. وجود وقفه در پردازندهها سیستم عامل دا از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. میستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Time-Driven هستند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel کند.	عبارت	
اجرا دستورات بر روی پردازنده بر عهده سیستم عامل است. سیستم عامل چیزی جز یک برنامه طولانی نیست که مدام کارهای سیستم را مدیریت می کند. کنترل و مدیریت نحوه اجرای برنامههای کاربر تماماً توسط سیستم عامل انجام می شود. کامپیوترهایی که DMA دارند، از سرعت بیشتری هنگام اجرای برنامهها برخوردارند. مدیریت سیستم فایل و مدیریت کاربران از وظایف سیستم عامل است. کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing) همان کامپیوترهای پردازش سنگین در حال چک کردن هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. است. است. است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستمهای عامل قدیمی بیشتر Time-Driven هستند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند.		نادرست؟
سیستم عامل چیزی جز یک برنامه طولانی نیست که مدام کارهای سیستم را مدیریت می کند. کنترل و مدیریت نحوه اجرای برنامههای کاربر تماماً توسط سیستم عامل انجام می شود. کامپیوترهایی که DMA دارند، از سرعت بیشتری هنگام اجرای برنامهها برخوردارند. مدیریت سیستم فایل و مدیریت کاربران از وظایف سیستم عامل است. کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing) هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. با در نظر گرفتن سهم سیستم عامل، همیشه میزان بهرهگیری از CPU Utilization) CPU با در حال اجرا است. زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل اجباری است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. وجود وقفه در پردازندهها بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای OI انجام میدهند. سیستمهای عامل امروزی بیشتر Time-Driven هستند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Time-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اخد.	کامپیوترها بدون سیستمعامل هم میتوانند کار کنند.	
کنترل و مدیریت نحوه اجرای برنامههای کاربر تماماً توسط سیستم عامل انجام می شود. کامپیوترهایی که DMA دارند، از سرعت بیشتری هنگام اجرای برنامهها برخوردارند. مدیریت سیستم فایل و مدیریت کاربران از وظایف سیستم عامل است. کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing) هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. است. با در نظر گرفتن سهم سیستم عامل، همیشه میزان بهرهگیری از CPU Utilization) (CPU با در حال اجرا است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند.	اجرا دستورات بر روی پردازنده بر عهده سیستم عامل است.	
کامپیوترهایی که DMA دارند، از سرعت بیشتری هنگام اجرای برنامهها برخوردارند. مدیریت سیستم فایل و مدیریت کاربران از وظایف سیستم عامل است. کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing) هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. با در نظر گرفتن سهم سیستمعامل، همیشه میزان بهرهگیری از CPU Utilization) CPU درصد است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستمعامل اجباری است. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند.	سیستم عامل چیزی جز یک برنامه طولانی نیست که مدام کارهای سیستم را مدیریت می کند.	
مدیریت سیستم فایل و مدیریت کاربران از وظایف سیستم عامل است. کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing) هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. است. با در نظر گرفتن سهم سیستم عامل، همیشه میزان بهرهگیری از CPU Utilization) (CPU) ۱۰۰ درصد است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل اجباری است. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستمهای عامل قدیمی بیشتر Time-Driven بیشتر این سیستمهای عامل امروزی بیشتر Time-Driven هستند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اخبار کند.	کنترل و مدیریت نحوه اجرای برنامههای کاربر تماماً توسط سیستم عامل انجام میشود.	
کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing) هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. است. با در نظر گرفتن سهم سیستمعامل، همیشه میزان بهرهگیری از CPU Utilization) CPU درصد است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستمعامل اجباری است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر میکند. امروز سیستمهاملها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام میدهند. سیستمهای عامل آمروزی بیشتر Time-Driven بودند. سیستمهای عامل آمروزی بیشتر Event-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند.	کامپیوترهایی که DMA دارند، از سرعت بیشتری هنگام اجرای برنامهها برخوردارند.	
هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. با در نظر گرفتن سهم سیستم عامل، همیشه میزان بهره گیری از CPU Utilization) ۲۰۰ درصد است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل اجباری است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عامل ها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام می دهند. سیستم های عامل قدیمی بیشتر Time-Driven بودند. سیستم های عامل امروزی بیشتر Event-Driven هستند. سیستم های عامل امروزی بیشتر Interrupt-Driven هستند. سیستم های عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند.	مدیریت سیستم فایل و مدیریت کاربران از وظایف سیستم عامل است.	
هستند. در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن است. با در نظر گرفتن سهم سیستم عامل، همیشه میزان بهره گیری از CPU Utilization) ۲۰۰ درصد است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل اجباری است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عامل ها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام می دهند. سیستم های عامل قدیمی بیشتر Time-Driven بودند. سیستم های عامل امروزی بیشتر Event-Driven هستند. سیستم های عامل امروزی بیشتر Interrupt-Driven هستند. سیستم های عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند.	کامپیوترهای انبوهی (Batch computers) همان کامپیوترهای پردازش سنگین (High computing)	
است. با در نظر گرفتن سهم سیستم عامل، همیشه میزان بهره گیری از CPU Utilization) CPU درصد است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل اجباری است. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عامل ها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای I انجام می دهند. سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند.	هستند.	
با در نظر گرفتن سهم سیستم عامل، همیشه میزان بهره گیری از CPU Utilization) ۲۰۰ درصد است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل اجباری است. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عامل ها فقط بر اساس وقفه ها هستند زیرا وقفه ها مدیریت دستگاههای IO انجام می دهند. سیستم های عامل قدیمی بیشتر Time-Driven بودند. سیستم های عامل امروزی بیشتر Event-Driven هستند. سیستم های عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	در زمانی که برنامههای کاربر وجود ندارد، سیستم عامل بصورت دایم در حلقه بینهایت در حال چک کردن	
است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستمهای عامل ها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام می دهند. سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	است.	
است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عاملها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام می دهند. سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	با در نظر گرفتن سهم سیستمعامل، همیشه میزان بهرهگیری از CPU Utilization) CPU درصد	
است. وجود وقفه در پردازندهها برای سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عاملها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام می دهند. سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	است، زیرا در زمانی که برنامه کاربر نیست، این سیستم عامل است که داخل حلقه فون-نیومن در حال اجرا	
وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عاملها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام می دهند. سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.		
وجود وقفه در پردازندهها، سیستم عامل را از اجرای کارهای کاربر و دیگر فعالیتها باخبر می کند. امروز سیستم عاملها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام می دهند. سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولید کنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	وجود وقفه در پردازندهها برای سیستمعامل اجباری است.	
سیستمهای عامل قدیمی بیشتر Time-Driven بودند. سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.		
سیستمهای عامل امروزی بیشتر Event-Driven هستند. سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	امروز سیستمعاملها فقط بر اساس وقفهها هستند زیرا وقفهها مدیریت دستگاههای IO انجام میدهند.	
سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	سیستمهای عامل قدیمی بیشتر Time-Driven بودند.	
سیستمهای عامل امروزی بیشتر Interrupt-Driven هستند. سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.	سیستمهای عامل امروزی بیشتر Event-Driven هستند.	
سیستمهای عامل بدون قابلیتهای خاص خود که باید توسط تولیدکنندگان پردازنده ایجاد شود، قادر به کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.		
کنترل تمام اتفاقات داخل کامپیوتر نیستند. روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.		
روشی وجود دارد که کاربر قادر است که برنامه خود را در حالت Kernel اجرا کند.		

تغییر حالت از Wer mode به Wer mode بصورت سختافزاری انجام می شود. تغییر حالت از User mode به Mer mode بصورت نرمافزاری انجام می شود. تغییر حالت از Wer mode به Wer mode بصورت سختافزاری انجام می شود. اگر از دستور IRET در انتهای توابع کاربر استفاده شود، ایرادی پیش نمی آید. اگر از دستور RET در انتهای برنامههای ISR استفاده شود، ایرادی پیش نمی آید. وچنانچه برنامه اسمبلی بنویسیم و در آن دستور (ایدن نیش نمی آید. تگه داشته و مسیر اجرایی به سیستم عامل برنگردد. آن بخش از هسته سیستم عامل که همواره در حافظه مقیم است، فقط ISR هاست. تعویض متن (Context switch) تماماً نرمافزاری انجام می شود. وقفه تودرتو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. وزاخوانی توابع تودرتو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام می شود. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، نرمافزاری انجام می شود.		
تغییر حالت از Wer mode به Wernel mode بصورت سختافزاری انجام می شود. اگر از دستور RET در انتهای توابع کاربر استفاده شود، ایرادی پیش نمی آید. اگر از دستور RET در انتهای برنامههای h ISR استفاده شود، ایرادی پیش نمی آید. چنانچه برنامه اسمبلی بنویسیم و در آن دستور () Exit نگذاریم، میتوانیم همیشه برنامه را در حافظه مقیم نگه داشته و مسیر اجرایی به سیستم عامل برنگردد. آن بخش از هسته سیستمعامل که همواره در حافظه مقیم است، فقط ISR هاست. تعویض متن (Context switch) تماماً نرمافزاری انجام می شود. تعویض متن (Context switch) تماماً سختافزاری انجام می شود. وقفه تودرتو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام می شود.	تغییر حالت از User mode به Kernel mode بصورت سختافزاری انجام می شود.	
اگر از دستور RET در انتهای توابع کاربر استفاده شود، ایرادی پیش نمی آید. اگر از دستور RET در انتهای برنامههای ISR استفاده شود، ایرادی پیش نمی آید. چنانچه برنامه اسمبلی بنویسیم و در آن دستور ()Exit نگذاریم، میتوانیم همیشه برنامه را در حافظه مقیم نگه داشته و مسیر اجرایی به سیستم عامل برنگردد. آن بخش از هسته سیستم عامل که همواره در حافظه مقیم است، فقط ISR هاست. تعویض متن (Context switch) تماماً نرمافزاری انجام می شود. تعویض متن (Context switch) تماماً سختافزاری انجام می شود. وقفه تودرتو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام می شود.	تغییر حالت از Kernel mode به User mode بصورت نرمافزاری انجام می شود.	
اگر از دستور RET در انتهای برنامههای Exit() ایند. چنانچه برنامه اسمبلی بنویسیم و در آن دستور ()Exit نگذاریم، میتوانیم همیشه برنامه را در حافظه مقیم نگه داشته و مسیر اجرایی به سیستم عامل برنگردد. آن بخش از هسته سیستمعامل که همواره در حافظه مقیم است، فقط ISR هاست. تعویض متن (Context switch) تماماً نرمافزاری انجام می شود. تعویض متن (Context switch) تماماً سختافزاری انجام می شود. وقفه تودرتو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام می شود.	تغییر حالت از Kernel mode به User mode بصورت سختافزاری انجام می شود.	
چنانچه برنامه اسمبلی بنویسیم و در آن دستور (Exit) نگذاریم، میتوانیم همیشه برنامه را در حافظه مقیم نگه داشته و مسیر اجرایی به سیستم عامل برنگردد. آن بخش از هسته سیستم عامل که همواره در حافظه مقیم است، فقط ISR هاست. تعویض متن (Context switch) تماماً نرمافزاری انجام میشود. تعویض متن (Context switch) تماماً سختافزاری انجام میشود. وقفه تودرتو موجب کندی سیستم میشود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم میشود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام میشود.	اگر از دستور iRET در انتهای توابع کاربر استفاده شود، ایرادی پیش نمی آید.	
نگه داشته و مسیر اجرایی به سیستم عامل برنگردد. آن بخش از هسته سیستم عامل که همواره در حافظه مقیم است، فقط ISR هاست. تعویض متن (Context switch) تماماً نرمافزاری انجام می شود. تعویض متن (Context switch) تماماً سخت افزاری انجام می شود. وقفه تودرتو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سخت افزاری انجام می شود.	اگر از دستور RET در انتهای برنامههای ISR hستفاده شود، ایرادی پیش نمی آید.	
آن بخش از هسته سیستمعامل که همواره در حافظه مقیم است، فقط ISR هاست. تعویض متن (Context switch) تماماً نرمافزاری انجام می شود. تعویض متن (Context switch) تماماً سختافزاری انجام می شود. وقفه تودرتو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام می شود.	چنانچه برنامه اسمبلی بنویسیم و در آن دستور ()Exit نگذاریم، میتوانیم همیشه برنامه را در حافظه مقیم	
تعویض متن (Context switch) تماماً نرمافزاری انجام می شود. تعویض متن (Context switch) تماماً سخت افزاری انجام می شود. وقفه تودر تو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودر تو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایه های زیرین به بالا در سلسله مراتب حافظه، سخت افزاری انجام می شود.	نگه داشته و مسیر اجرایی به سیستم عامل برنگردد.	
تعویض متن (Context switch) تماماً سختافزاری انجام می شود. وقفه تودرتو موجب کندی سیستم می شود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم می شود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایه های زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام می شود.	آن بخش از هسته سیستمعامل که همواره در حافظه مقیم است، فقط ISR هاست.	
وقفه تودرتو موجب کندی سیستم میشود در نتیجه باید جلوی آنرا گرفت. فراخوانی توابع تودرتو موجب کندی سیستم میشود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام میشود.	تعویض متن (Context switch) تماماً نرمافزاری انجام می شود.	
فراخوانی توابع تودرتو موجب کندی سیستم میشود، در نتیجه باید جلوی آنرا گرفت. آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام میشود.	تعویض متن (Context switch) تماماً سختافزاری انجام می شود.	
آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام میشود.	وقفه تودرتو موجب کندی سیستم میشود در نتیجه باید جلوی آنرا گرفت.	
	فراخوانی توابع تودرتو موجب کندی سیستم میشود، در نتیجه باید جلوی آنرا گرفت.	
آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، نرمافزاری انجام میشود.	آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، سختافزاری انجام میشود.	
	آوردن اطلاعات از لایههای زیرین به بالا در سلسله مراتب حافظه، نرمافزاری انجام میشود.	

۲– موارد زیر را تعریف کنید:

توضيح	مفهوم	ترجمه فارسی
	Boot	
	Bootstrap	
	Firmware	
	Usability	
	Workstation	
	Mainframe	
	Interrupt	
	Vector	
	Trap /Exception	
	Polling	
	Nonvolatile	
	memory	
	Solid state disk	
	Throughput	
	Asymmetric multiprocessors	

Blade
processor
Storage Area
Network
Distributed
Lock Manager
Job pool
Race condition

٣- به سوالات زير پاسخ دهيد:

الف) فرق Cache با Buffer چیست؟

دكامپيوتريها چيست؟	ا) نسىت بە جند	ندهها (با حندهستهای،ه	وزافزون حندير دازا	ب) سه دلیل افزایش ر
	* * * `	U	, ,, ,, ,, ,, ,, ,, ,, ,, ,, ,, ,, ,, ,	,

توضيح	دلیل

ج) فرق حافظههای UMA با NUMA چیست و کدامیک بهتر هستند؟

د) عامل اصلی در کندی افزایش تعداد هستهها در پردازندههای چند هستهای چیست؟

ه) یک مثال از کامپیوترهای جفت شده سفت (Tightly Coupled) و یک مثال از کامپیوترهای جفت شده شل (Loosely کی مثال از کامپیوترهای جفت شده شل (Coupled) بیاورید.

Loosely Coupled	Tightly Coupled

و) دنیای محاسبات تعاملی چطور محاسباتی است؟ و سیستم عامل برای این دنیای محاسباتی چطور باید باشد؟

ز) زمانبند مراقب چیست و عمده استفاده آن چیست؟

ح) فرق محاسبات توزيعي (Distributed Computing) با محاسبات شبكهاي (Network Computing) چيست؟

شتری-خدمتگذار (Client Server)؟	ط) سیستمهای نظیر –به –نظیر (Peer-to-Peer) بهتر هستند یا م
Client-S بیاورید.	ی) یک مثال رایج از سیستم Peer-to-Peer و یک مثال از erver
· · · · · · · · · · · · · · · · · · ·	Peer-to-Peer
	Client-Server
بیست؟	ک) فرق تقلید (Emulation) با مجازیسازی (Virtualization) چ
ام ببرید. (۳ مورد)	ر) مهمترین سرویسها در محاسبات ابری را همراه با مثال فقط ن
ـد نوع هستند؟	م) سیستمهای بیدرنگ چیستند؟ با ذکر مثال توضیح دهید چن
	ن) سیستمهای نهفته چیستند؟ با ذکر مثال توضیح دهید.
System prog هستند در قیاس با	س) مزایا و معایب سرویسهای سیستم عامل که بصورت ram; کر کنید.
عيب	مزیت
تم عامل را نام برده و با هم مقایسه کنید.	ع) دو ارتباط بین فر آیندها (برای تبادل اطلاعات) از طریق سیس
ارتباط ۲:	ارتباط ۱:
	معایب
	مزایا

ف) ساختارهای تولید سیستمعامل را در قالب جدول زیر تکمیل کنید:

مثال سیستم عامل	ذکر یک عیب	مهمترین مشخصه	ساختار	معادل فارسی
			Monolithic	
			Layered	
			Microkernel	
			Modules	
			Hybrid	