İşlemler

Dr. Günay TEMÜR

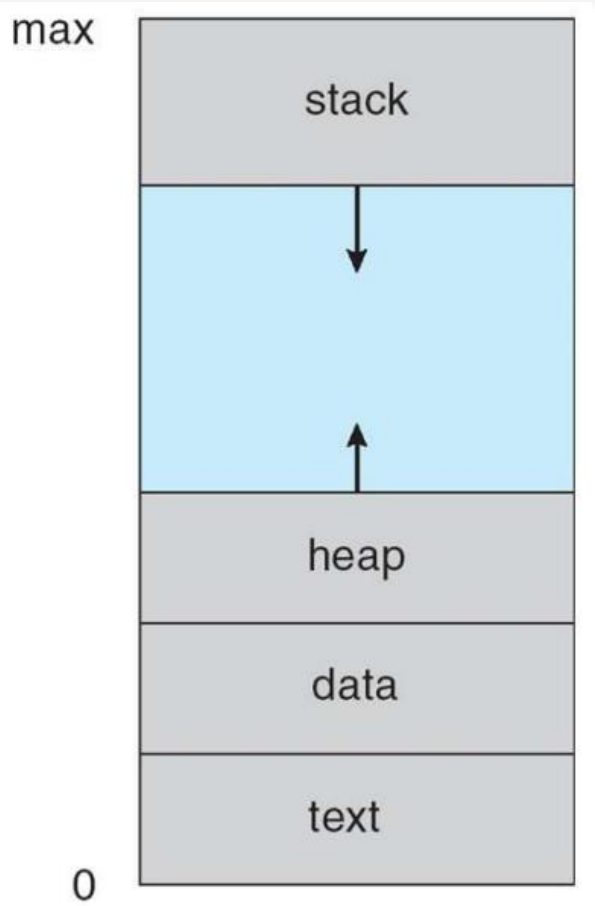
Bölüm 3: İşlemler

- İşlem Kavramı
- İşlem Zamanlaması (Process Scheduling)
- İşlemler Üzerindeki Faaliyetler
- İşlemler Arası İletişim (Interprocess Communication)
- IPC Sistemi Örnekleri
- İstemci-Sunucu Sistemlerde İletişim

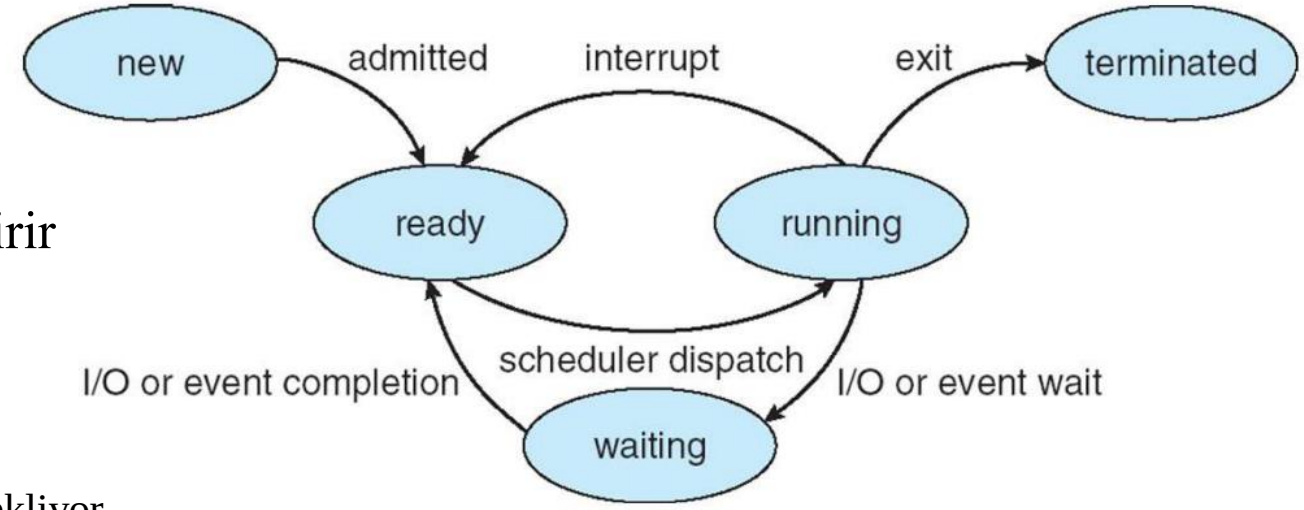
İşlem Kavramı

- İşletim sistemleri farklı tipte programlar çalıştırabilir:
  - Toplu iş sistemleri - işler
  - Zaman paylaşımlı sistemler (time-shared systems) – kullanıcı programları veya görevleri
- Ders kitabı iş (job) ve işlem (process) kelimelerini kabaca birbirleri yerine kullanabilmekte
- İşlem – çalışmakta olan bir programdır
- İşlemler aşağıdakileri içermelidir:
  - program sayacı (program counter)
  - yığın (stack)
  - veri bölümü (data section)
  - Heap (süreç sırası)

Hafızadaki İşlemler

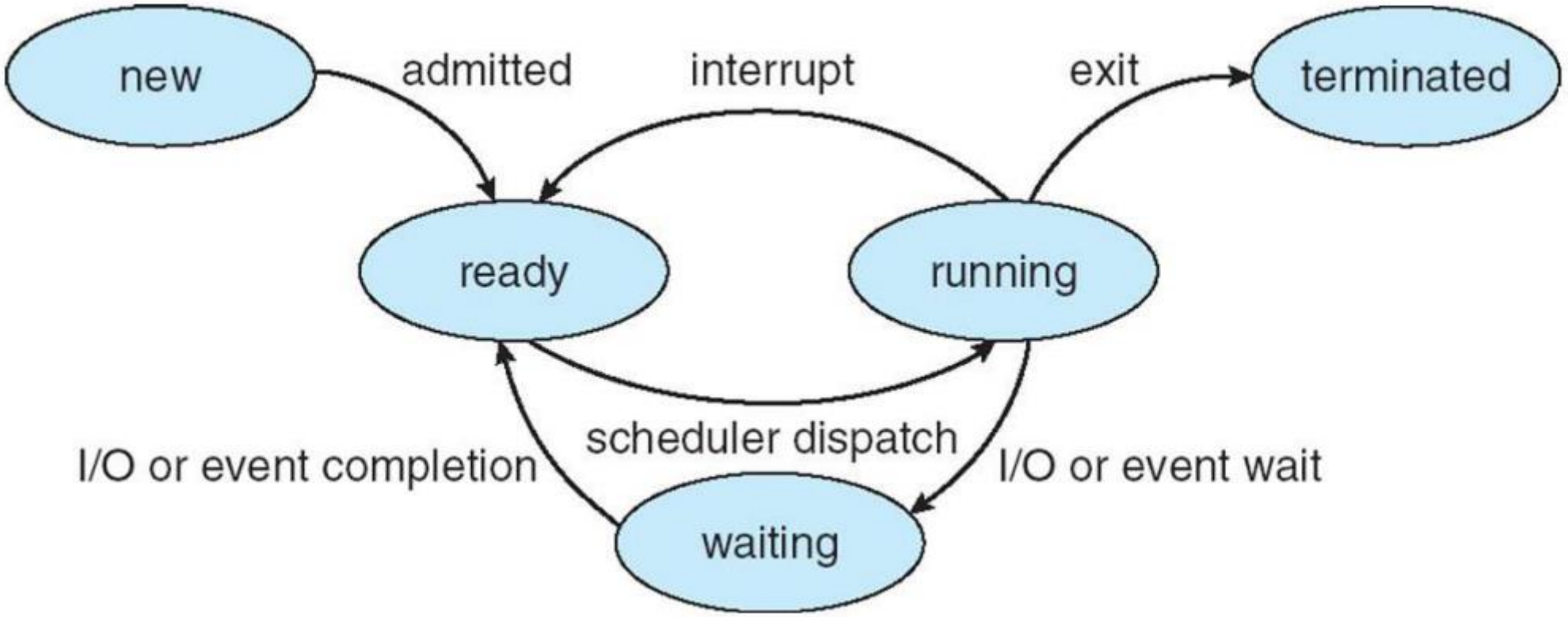


İşlem Durumu



- Bir işlem çalıştırılırken durumunu (state) değiştirir
  - yeni (new): İşlem oluşturuldu
  - çalışıyor (running): İşlem komutları çalıştırılıyor
  - bekliyor (waiting): İşlem bir olayın gerçekleşmesini bekliyor
  - hazır (ready): İşlem bir işlemciye atanmayı bekliyor
  - sonlandırılmış (terminated): İşlem çalışmayı bitirmiş

İşlem Durumları Akış Şeması

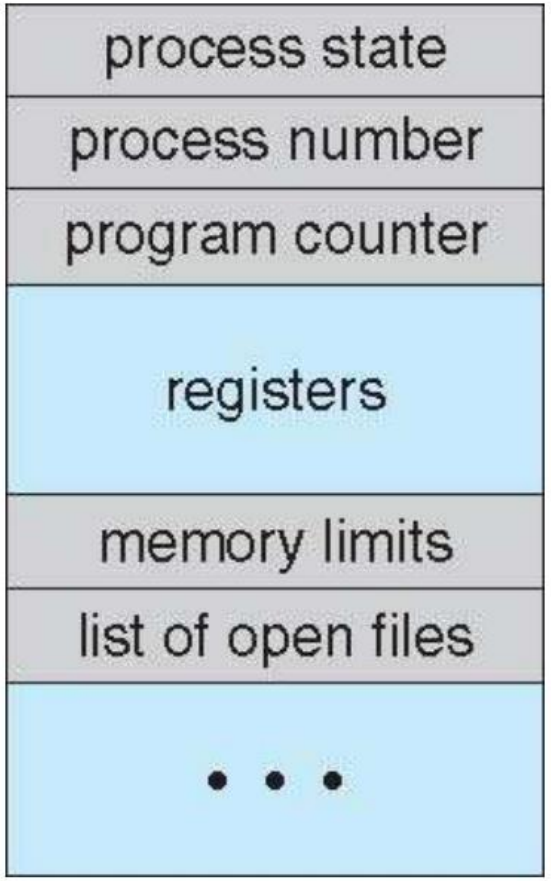


İşlem Kontrol Bloğu (PCB)

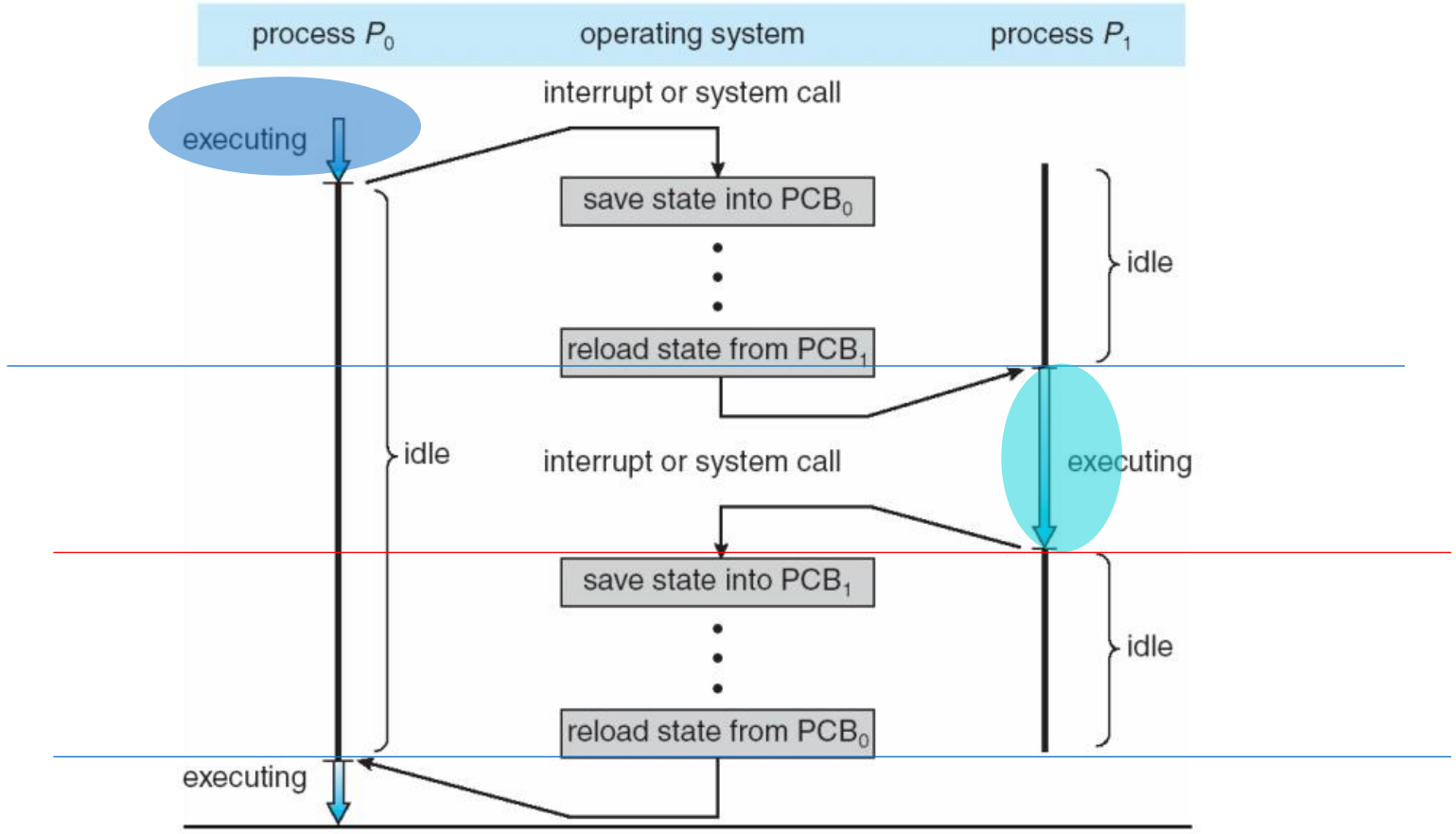
Herhangi bir işlem ile ilişkili bilgiler (process control block)

- İşlem durumu
- İşlem sayacı
- CPU yazmaçları (CPU registers)
- CPU zamanlama bilgileri
- Hafıza yönetim bilgileri
- Hesap (accounting) bilgileri
- I/O durum bilgileri

İşlem Kontrol Bloğu (PCB)



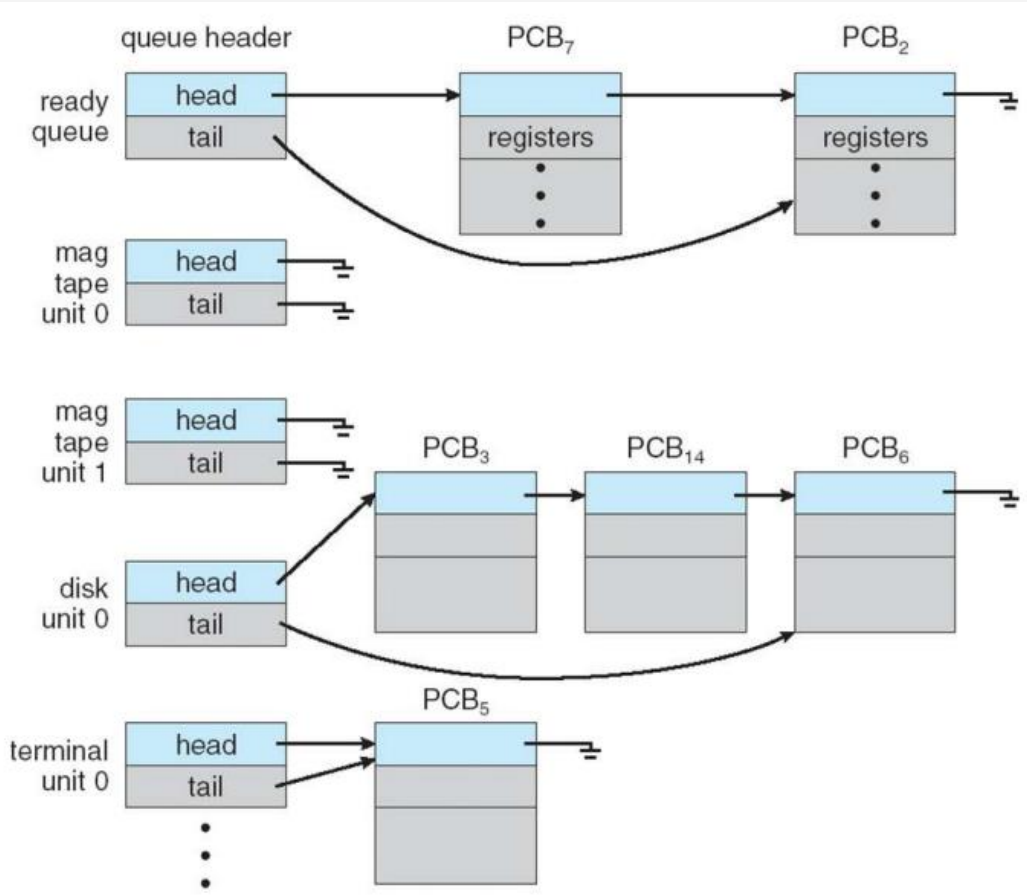
CPU İşlemden İşleme Geçiş Yapar



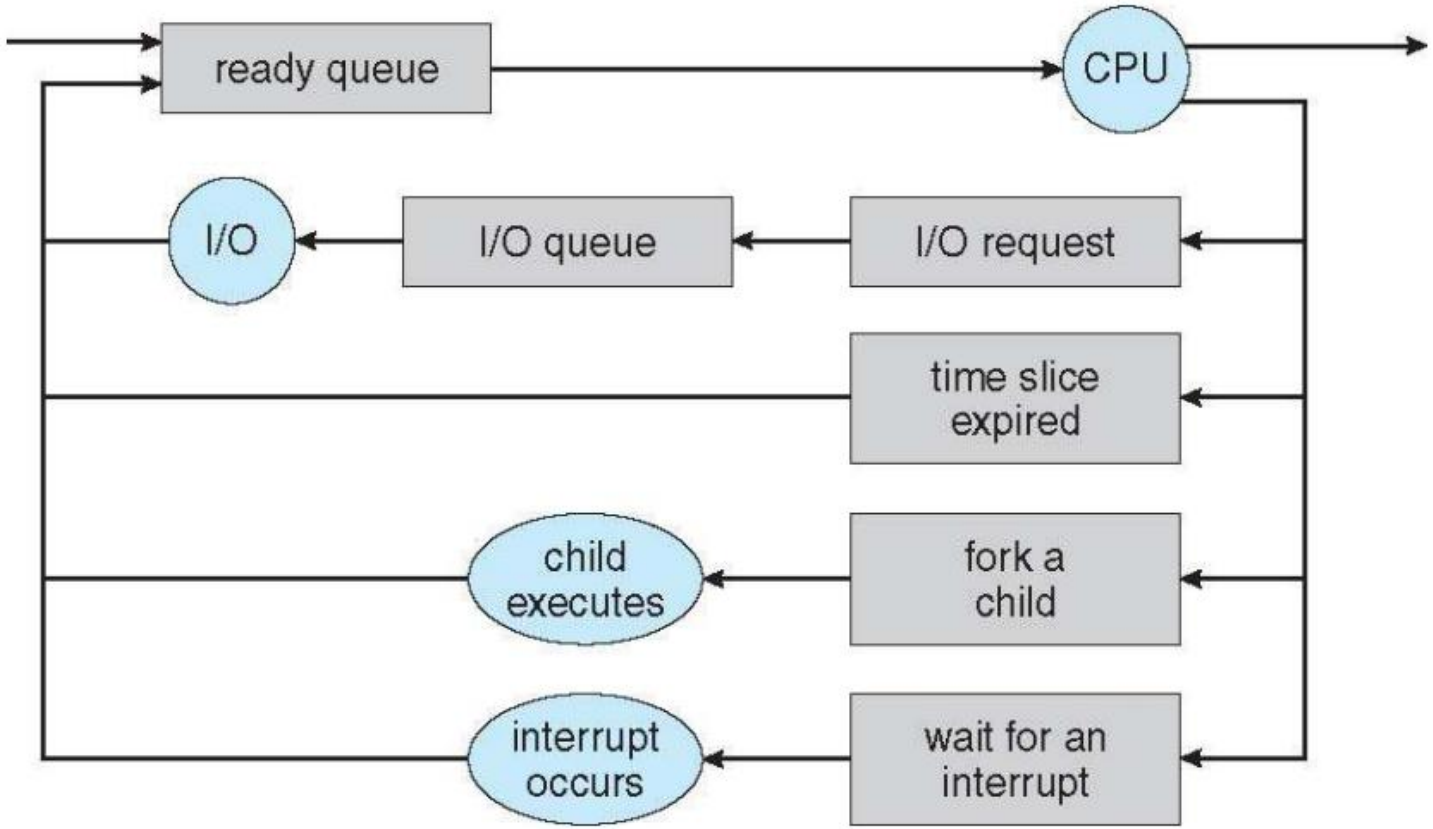
İşlem Zamanlama Kuyrukları

- Process Scheduling Queues
- İş kuyruğu (job queue) – sistemdeki tüm işlemlerin kümesi
- Hazır kuyruğu (ready queue) – ana hafızadaki, tüm işlemlerin kümesi - çalışmaya hazır ve çalıştırılmayı bekleyen
- Cihaz kuyrukları (device queues) – bir I/O cihazını kullanmayı bekleyen işlemler kümesi
- İşlemler değişik kuyruklar arasında geçiş yapar

Hazır Kuyruğu ve Bazı I/O Cihaz Kuyrukları



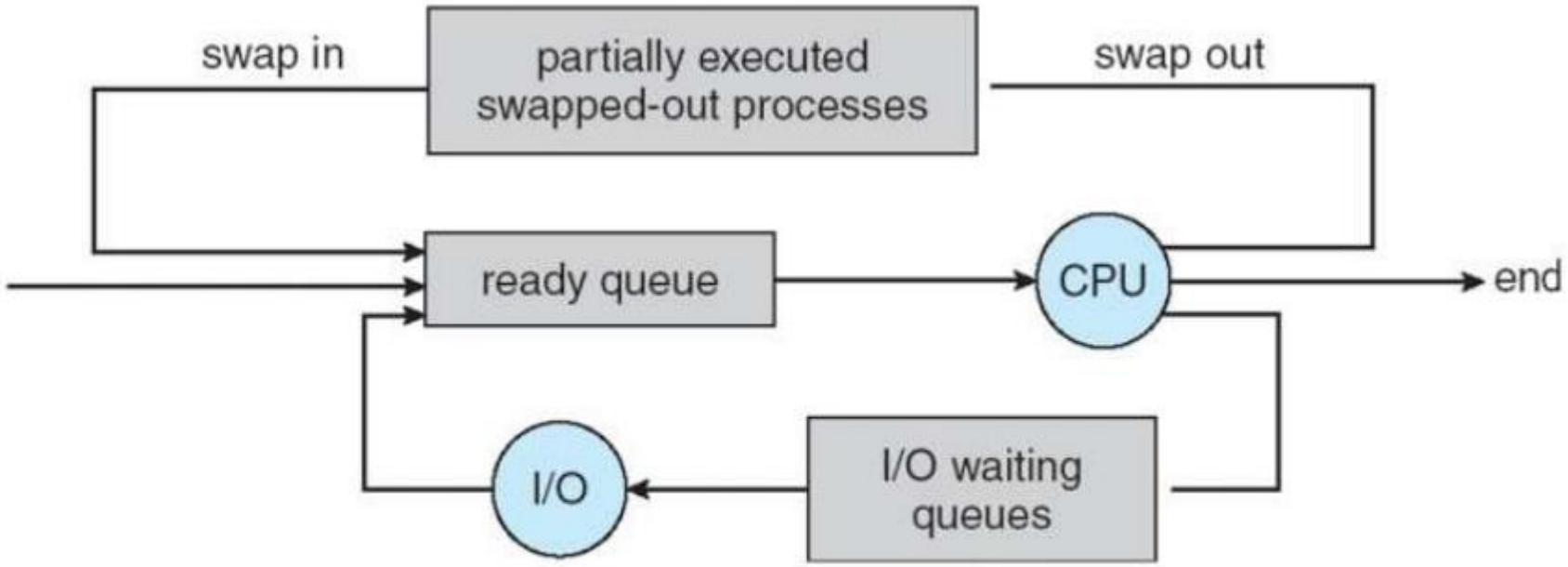
İşlem Zamanlaması



Zamanlayıcılar

- Uzun vadeli zamanlayıcılar (long-term scheduler)
  - veya iş zamanlayıcısı (job scheduler)
  - hangi işlemlerin hazır kuyruğuna (ready queue) alınması gerektiğine karar verir
- Kısa vadeli zamanlayıcılar (short-term scheduler)
  - veya CPU zamanlayıcısı
  - sıradaki hangi işlemin CPU tarafından çalıştırılacağına karar verir

Orta Vadeli Zamanlama



Orta Vadeli Zamanlayıcı (medium-term schedular)

Zamanlayıcılar

- Kısa vadeli zamanlayıcı çok sık çalıştırılır (milisaniye) hızlı çalışmalı
- Uzun vadeli zamanlayıcı çok sık çalıştırılmaz (saniye, dakika) yavaş olabilir
- Uzun vadeli zamanlayıcı çoklu programlamanın (multiprogramming) seviyesini kontrol eder
- İşlemler aşağıdaki iki kategoriye ayrılabilir:
  - I/O ağırlıklı işlemler (I/O-bound process) – CPU üzerinde kısa süreli işlerler yaparlar ve zamanlarının çoğu I/O işleri yaparak geçer
  - CPU ağırlıklı işlemler (CPU-bound process) – zamanlarının çoğunu CPU üzerinde uzun işlemler yaparak geçirirler, I/O işleri çok daha azdır

Ortam Değişikliği

- Context Switch
- CPU başka bir işleme geçerken, sistem eski işlemin durumunu kaydetmeli ve yeni işlemin kaydedilmiş durumunu yüklemelidir
- Bir işlemin ortamı (context) PCB'de tutulur
- Ortam değişikliği için harcanan zaman ek yüktür. Değişim sırasında işlemci kullanıcının işine direk yarayan bir iş yapmamaktadır
- Harcanan zaman donanım desteğine bağlıdır

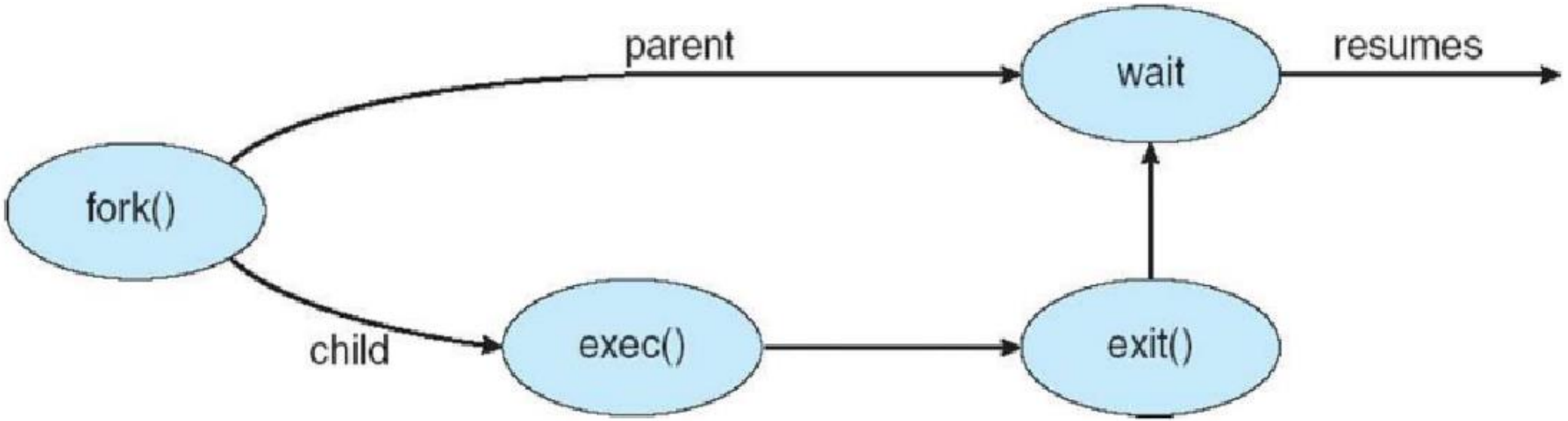
İşlem Oluşturma

- Ana işlem (parent process) çocuk işlemleri (children processes) oluşturur
- Çocuk işlemlerde yeni işlem oluşturabileceğinden, sistemde işlemlerin bir ağacı oluşur
- Genellikle işlemler işlem belirteci (process identifier - pid) kullanılarak birbirlerinden ayrılırlar ve yönetilirler
- Kaynak paylaşımı
  - Ana işlem ve çocuklar tüm kaynakları paylaşırlar
  - Çocuklar ana işlemin kaynaklarının belli bir alt kümesini paylaşır
  - Ana işlem ve çocuklar kaynak paylaşmazlar
- Çalıştırma
  - Ana işlem ve çocuklar aynı anda çalışır
  - Ana işlem, çocuk işlemler sonlanana kadar bekler

İşlem Oluşturma

- Hafıza alanı (address space)
  - Çocuk ana işlemin kopyasına sahip
  - Çocuk adres alanına yeni bir program yükler
- UNIX örnekleri
  - fork sistem çağrısı yeni bir işlem oluşturur
  - exec sistem çağrısı, fork sistem çağrısı ile yeni işlem oluşturulduktan sonra, işlemin hafıza alanına yeni bir program yüklemek için kullanılır

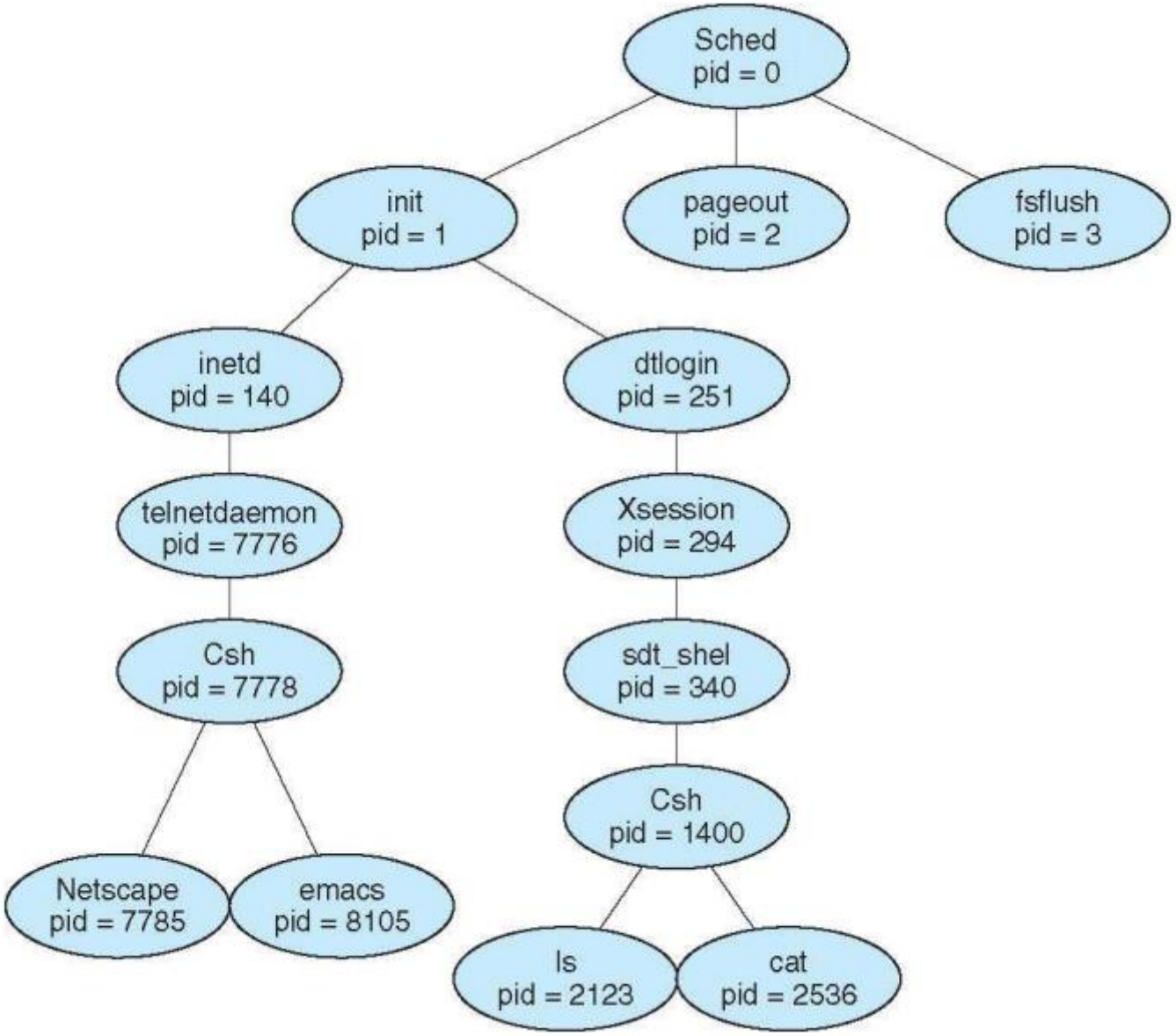
İşlem Oluşturma



Yeni İşlem Oluşturma

```
int main()
{
    pid_t pid;
    /* fork another process */
    pid = fork();
    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        exit(-1);
    }
    else if (pid == 0) { /* child process */
        execlp("/bin/ls", "ls", NULL);
    }
    else { /* parent process */
    /* parent will wait for the child to complete */
        wait(NULL);
        printf("Child Complete");
        exit(0);
    }
}
```

Soloris'te Tipik Bir İşlem Ağacı



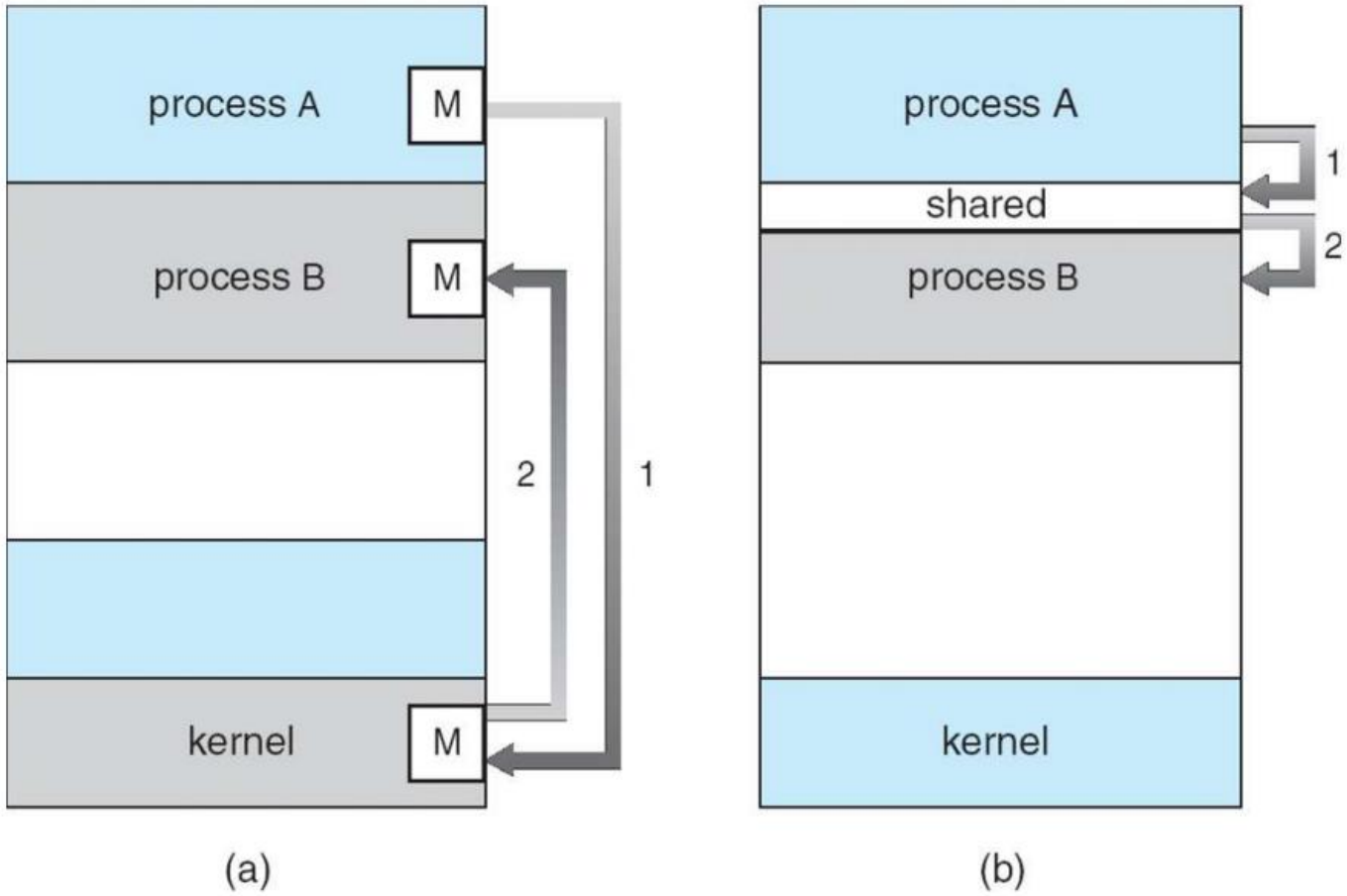
İşlem Sonlandırma

- Process Termination
- İşlemler son komutlarını çalıştırdıktan sonra işletim sistemine kendilerini silmelerini isterler (exit)
  - İşlemin dönüş değeri çocuktan ana işleme döndürülür (wait ile)
  - İşlemin kaynakları işletim sistemi tarafından geri alınır
- Ana işlem çocuk işlemlerin çalışmasını sonlandırabilir (abort)
  - Çocuk kendine ayrılan kaynakları aşmışsa
  - Çocuk işleme atanan göreve artık ihtiyaç yoksa
  - Eğer ana işlem sonlandırılıyorsa
    - Bazı işletim sistemleri ana işlem sonlandırıldıktan sonra çocuklarının çalışmaya devam etmesine izin vermez
      - – Peşpeşe sonlandırma (cascading termination) ile tüm çocuk işlemler sonlandırılır

İşlemler Arası İletişim

- Interprocess Communication
- Bir sistemdeki işlemler bağımsız (independent) ya da işbirliği yapıyor (cooperating) olabilir
- İşbirliği yapan işlemler birbirlerini etkileyebilirler veya veri paylaşabilirler
- Neden işlemler işbirliği yapar?
  - Bilgi paylaşımı
  - İşlem hızını arttırmak
  - Modülerlik sağlamak
- İşbirliği yapan işlemler işlemler arası iletişime (interprocess communication - IPC) ihtiyaç duyar
- IPC'nin iki modeli
  - Paylaşımlı bellek (shared memory)
  - Mesaj gönderme (message passing)

İletişim Modelleri



Üretici-Tüketici Problemi

- Producer-Consumer Problem
- İşbirliği yapan işlemler için problem örneği: üretici (producer) işlem bilgi üretirken, tüketici (consumer) işlem üretilen bilgiyi tüketir
  - sınırsız tampon bellek (unbounded-buffer): tampon bellek için herhangi bir pratik sınır getirmez
  - sınırlı tampon bellek (bounded-buffer): sınırlı boyutta bir tampon bellek kullanıldığını varsayar

İşlemler Arası İletişim – Mesaj Gönderme

- İşlemlerin iletişim kurması ve eylemlerini senkronize etmelerini sağlayan mekanizma
- Mesaj sistemi – işlemler birbirleriyle paylaşımlı değişkenlere bağlı kalmaksızın haberleşir
- IPC mekanizması iki işlemi sağlar:
  - send (mesaj gönderme) – mesaj boyutu sabit ya da değişken olabilir
  - receive (mesaj alma)
- Eğer P ve Q işlemleri iletişim kurmak isterse:
  - birbirleri arasında bir iletişim bağlantısı (communication link) sağlamalıdırlar
  - send/receive kullanarak mesajlaşmalıdırlar
- İletişim bağlantısının gerçekleştirimi
  - fiziksel (e.g., paylaşımlı hafıza, donanım hattı)
  - mantıksal (e.g., mantıksal özellikler)

BİTTİ