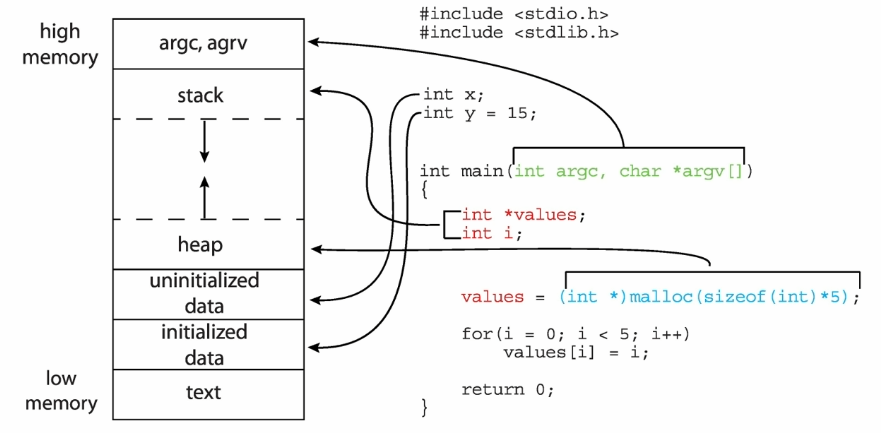
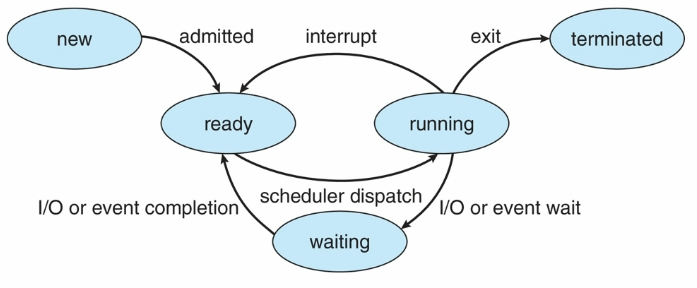
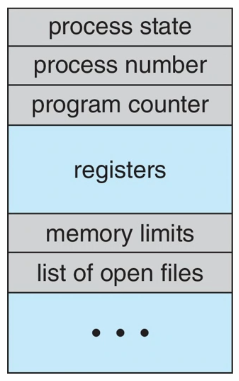
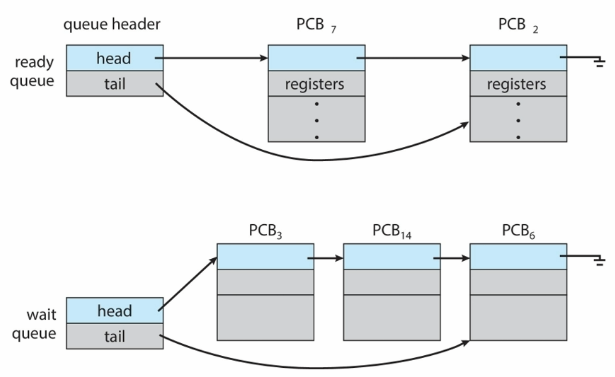
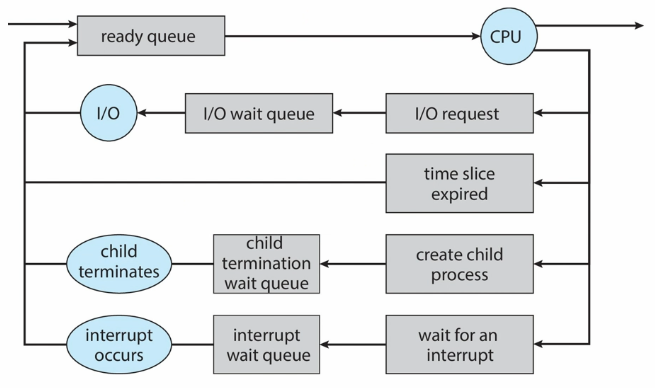
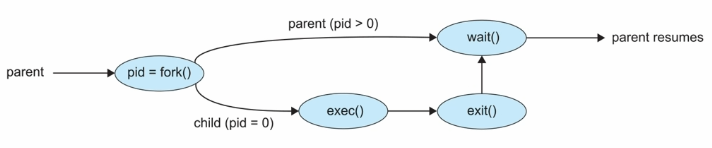
Chapter 3: Processes

1. **Process Concept; 프로세스 개념**
   1. **The Process; 프로세스**
      1. An Operating system executes a variety of programs that run as a process.  
         운영 체제는 프로세스로 실행되는 다양한 프로그램을 실행한다.
      2. Process – a program in execution; process execution must progress in sequential fashion  
         프로세스 – 실행 중인 프로그램으로, 프로세스 실행은 순차적인 방식으로 진행되어야 한다.
         1. 텍스트 섹션 – 실행 코드
         2. 데이터 섹션 – 전역 변수
         3. 힙 섹션 – 프로그램 실행 중에 동적으로 할당되는 메모리
         4. 스택 섹션 – 함수를 호출할 때 임시 데이터 저장장소(예: 함수 매개변수, 복귀 주소 및 지역 변수)
      3. Multiple parts
         1. The program code, also called text section  
            프로그램 코드를 텍스트 섹션이라 부른다
         2. Current activity including program counter, processor registers  
            프로세스의 현재 활동의 상태는 프로그램 카운터 값과 프로세서 레지스터의 내용으로 나타낸다.
         3. Stack containing temporary data  
            스택 섹션은 임시 데이터(예: 함수 매개변수, 복귀 주소 및 지역 변수)를 포함한다.
            1. Function parameter, return addresses, local variables  
               함수 매개변수, 복귀 주소 및 지역 변수
         4. Data section containing global variables  
            데이터 섹션은 전역 변수를 포함한다.
         5. Heap containing memory dynamically allocated during run time  
            힙 섹션은 프로그램 실행 중에 동적으로 할당되는 메모리를 포함한다.
      4. Program is passive entity stored on disk (executable file); process is active  
         프로그램은 디스크에 저장된 파일(실행 파일)과 같은 수동적인 존재, 프로세스는 능동적인 존재이다.
         1. Program becomes process when executable file loaded into memory  
            실행 파일이 메모리에 적재될 때 프로그램은 프로세스가 된다.
      5. Execution of program started via GUI mouse clicks, command line entry of its name, etc
      6. One program can be several processes
         1. Consider multiple users executing the same program
      7. Memory Layout of a C Program; C 프로그램의 메모리 배치  
         
   2. **Process State; 프로세스 상태**
      1. As a process executes, it changes state  
         프로세스가 실행되면, 그 상태가 변한다.
         1. **New**: The process is being created  
            새로운: 프로세스가 생성 중이다.
         2. **Running**: Instructions are being executed  
            실행: 명령어들이 실행되고 있다. by CPU
         3. **Waiting**: The process is waiting for some event to occur  
            대기: 프로세스가 어떤 이벤트(I/O 또는 신호의 수신 같은)가 일어나기를 기다린다.
         4. **Ready**: The process is waiting to be assigned to a processor  
            준비: 프로세스가 처리기(CPU)에 할당되기를 기다린다.
         5. **Terminated**: The process has finished execution  
            종료: 프로세스의 실행이 종료되었다.
         6. Diagram of Process State; 프로세스 상태 다이어그램  
            
   3. **Process Control Block (PCB); 프로세스 제어 블록**  
      Information associated with each process(also called task control block)  
      각 프로세스와 연관된 정보(태스크 제어 블록이라고도 불린다)
      1. Process state – running, waiting, etc  
         프로세스 상태 – 새로운(new), 준비(ready), 실행(running), 대기(waiting) 또는 정지(halted) 상태 등이다.
      2. Program counter – location of instruction to next execute  
         프로그램 카운터 – 다음에 실행할 명령어의 주소
      3. CPU registers – contents of all process-centric registers  
         CPU 레지스터들 – 모든 프로세스 중심 레지스터들의 내용
      4. CPU scheduling information – priorities, scheduling queue pointers  
         CPU 스케줄링 정보 – 프로세스 우선 순위, 스케줄 큐에 대한 포인터
      5. Memory-management information – memory allocated to the process  
         메모리 관리 정보 – 프로세스에 할당된 메모리
      6. Accounting information – CPU used, clock time elapsed since start, time limits  
         회계(accounting) 정보 – 사용된 CPU, 시작 이후 경과된 시간, 시간 제한
      7. I/O status information – I/O devices allocated to process, list of open files  
         입출력 상태 정보 – 프로세스에 할당된 입출력 장치, 열린 파일의 목록  
          
2. Process Scheduling; 프로세스 스케줄링  
   ・Maximize CPU use, quickly switch processes onto CPU core  
    CPU 이용을 최대화, CPU 코어를 빈번하게 교체  
   ・Process scheduler / Degree of Multiprogramming / IO bound process / CPU bound process  
    프로세스 스케줄러 / 다중 프로그래밍 정도 / I/O 바운드 프로세스 / CPU 바운드 프로세스  
   ・Process scheduler selects among available processes for next execution on CPU core  
    프로세서 스케줄러는 코어에서 실행 가능한 여러 프로세스 중에서 하나의 프로세스를 선택한다.
   1. Scheduling Queue; 스케줄링 큐
      1. Maintains scheduling queues of processes  
         프로세스의 스케줄링 큐 유지
         1. Ready queue – set of all processes residing in main memory, ready and waiting to execute  
            준비 큐 – 메인 메모리에 있는 프로세스, 실행되기를 준비하고 기다린다
         2. Wait queues – set of processes waiting for an event (i.e.I/O)  
            대기 큐 – 특정 이벤트(I/O 요청의 완료와 같은)가 발생하기를 기다리는 프로세스
         3. Processes migrate among the various queues
         4. Ready and Wait Queues  
            
         5. Representation of Process Scheduling; 프로세스 스케줄링을 나타내는 큐잉 다이어그램  
            
   2. Context Switch; 문맥 교환  
      A context switch occurs when the CPU switches from one process to another  
      문맥 교환은 CPU가 하나의 프로세스를 다른 프로세스로 전환할 때 발생한다.
      1. When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process via a context switch  
         CPU가 다른 프로세스로 전환할 때, 시스템은 반드시 이전 프로세스의 상태를 저장하고 문맥 교환을 통해 저장되어 있던 새 프로세스의 상태를 불러와야 한다.
      2. Context of a process represented in the PCB  
         PCB에서 표현된 프로세스의 문맥
      3. Context-switch time is overhead; the system does no useful work while switching  
         문맥 교환이 진행될 동안 시스템이 아무런 유용한 일을 못하기 때문에 문맥 교환 시간은 오버헤드이다.
         1. The more complex the OS and the PCB -> the longer the context switch  
            OS와 PCB가 복잡할수록 -> 문맥 교환 시간이 길어진다.
      4. Time dependent on hardware support  
         문맥 교환 시간은 하드웨어의 지원에 크게 좌우된다.
         1. Some hardware provides multiple sets of registers per CPU -> multiple contexts loaded at once  
            일부 하드웨어는 CPU당 여러 레지스터 집합을 제공한다. -> 여러 문맥을 한 번에 불러옴
3. Operations on Processes; 프로세스에 대한 연산
   1. Process Creation; 프로세스 생성
      1. Parent process create children processes, which, in turn create other processes, forming a tree of processes  
         부모 프로세스는 자식 프로세스를 생성하고, 차례로 다른 프로세스를 생성하여 프로세스 트리를 형성한다.
      2. Generally, process identified and managed via a process identifier (pid)  
         일반적으로 프로세스는 프로세스 식별자를 통해 식별 및 관리된다.
      3. Resource sharing options; 리소스 공유 옵션
         1. Parent and children share all resources  
            부모와 자식은 모든 리소스를 공유한다.
         2. Children share subset of parent’s resources  
            자녀는 부모의 리소스 일부를 공유한다.
         3. Parent and child share no resources  
            부모와 자녀가 리소스를 공유하지 않는다.
      4. Execution options; 실행 옵션
         1. Parent and children execute concurrently  
            부모와 자식이 동시에 실행
         2. Parent waits until children terminate  
            부모는 자식이 종료될 때까지 기다림
      5. Address space; 주소 공간
         1. Child duplicate of parent  
            자식은 부모의 복사본
         2. Child has a program loaded into it  
            자식이 자신에게 적재될 새로운 프로그램을 가지고 있다.
      6. UNIX examples
         1. fork() system call creates new process  
            새로운 프로세스는 fork() 시스템 콜로 생성된다.
         2. exec() system call used after a fork() to replace the process’ memory space with a new program  
            fork() 시스템 콜 다음에 exec() 시스템 콜을 사용하여 자신의 메모리 공간을 새로운 프로그램으로 교체한다.
         3. Parent process calls wait() for the child to terminate  
            부모는 wait() 시스템 콜로 자식 프로세스가 끝나길 기다린다.  
            
   2. Process Termination; 프로세스 종료
      1. Process executes last statement and then asks the operating system to delete it using the exit() system call.  
         프로세스가 마지막 문장의 실행을 끝내고, exit() 시스템 콜을 사용하여 운영체제에서 자신의 삭제를 요청하면 종료한다.
         1. Returns status data from child to parent (via wait())  
            프로세스는 부모 프로세스에 상태 값(통상 정수 값)을 반환할 수 있다. (wait() 시스템 콜을 통해)
         2. Process’ resources are deallocated by operating system  
            프로세스의 리소스는 할당 해제되고 운영체제로 반납된다.
      2. If no parent waiting (did not invoke wait()) process is a zombie  
         종료되었지만 부모 프로세스가 아직 wait() 호출을 하지 않은 프로세스를 좀비 프로세스라고 한다.
      3. If parent terminated without invoking wait, process is an orphan   
         부모 프로세스가 wait 호출 없이 종료된 프로세스를 고아 프로세스라고 한다.
4. Interprocess Communication; 프로세스 간 통신
   * 1. Processes within a system may be independent or cooperating  
        운영체제 내에서 실행되는 프로세스들은 독립적이거나 협력적인 프로세스들일 수 있다.
     2. Cooperating process can affect or be affected by other processes, including sharing data  
        협력적인 프로세스는 다른 프로세스들에 영향을 주거나 받을 수 있다. 데이터 공유를 포함해서
     3. Cooperating processes need interprocess communication (IPC)  
        협력적인 프로세스는 프로세스 간 통신(IPC)가 필요하다.
     4. Two models of IPC; IPC의 두 가지 모델
        1. Shared memory  
           공유 메모리
        2. Message passing  
           메시지 전달  
           