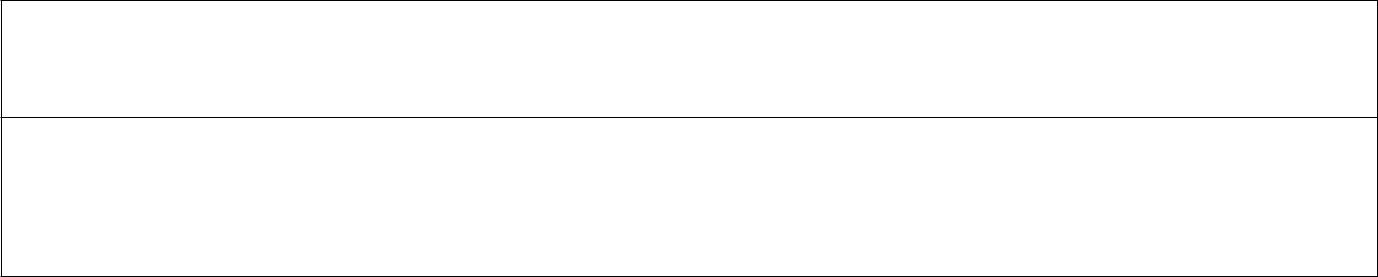
문제 1. <배점 : 4 × 5 = 20점> Fundamentals of OS

* Multitasking이 필요한 이유는 무엇이며, Multiprogramming과의 차이점은 무엇인지 간단하게 기술하시오.



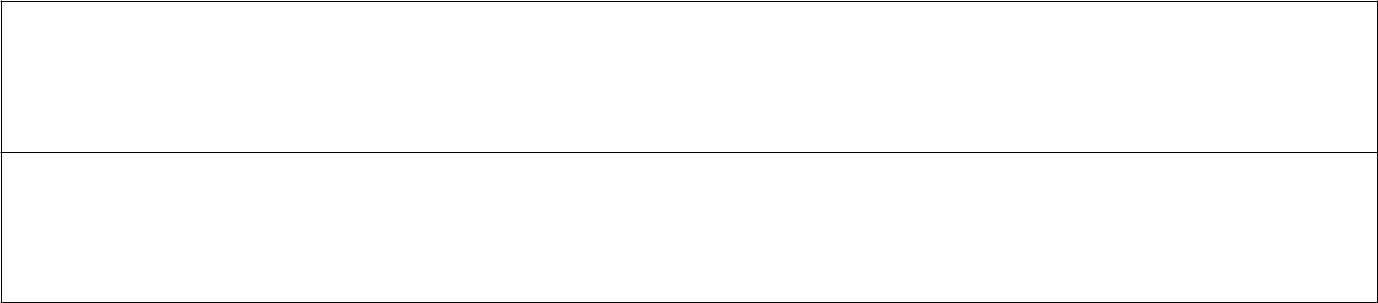
Multitasking이 필요한 이유 :

하나의 프로세스가 IO 작업을 수행할 때, CPU를 최대한 효율적으로 활용하기 위하여 다른 프로세스를 실행함.

Multiprogramming과의 차이점 :

하나의 프로세스가 CPU를 너무 오래 독점하는 것을 방지하기 위하여 (IO 작업을 수행하지 않는 경우라도) 일정한 시간 이 경과하면 프로세스 실행을 중단하고 다른 프로세스를 실행함

* Bootstrap이 수행하는 작업에는 어떠한 것들이 있으며, 이것이 저장되어 있는 매체의 특징을 간단하게 기술하시오.



Bootstrap이 수행하는 작업 :

시스템에 전원이 인가되었을 때, 하드웨어 초기화

운영체제를 하드디스크로부터 메모리로 적재하여 운영체제 부팅을 시작함

Bootstrap이 저장되어 있는 매체의 특징 :

전원이 없더라도 저장 내용이 보존되고, 그 내용이 변경되지 않도록 ROM에 저장됨.

이 ROM의 시작 주소는 CPU 내 PC의 초기값으로 설정되어야 함.

* Dual Mode가 필요한 이유와 이를 위하여 필요한 하드웨어 장치 및 이를 활용하는 방법을 간단하게 기술하시오.



Dual Mode가 필요한 이유 :

시스템의 하드웨어(등 중요한 자원들을) 보호하기 위하여, 운영체제에게만 접근을 허용하고 사용자가 직접 접근하는 것 을 방지하기 위함.

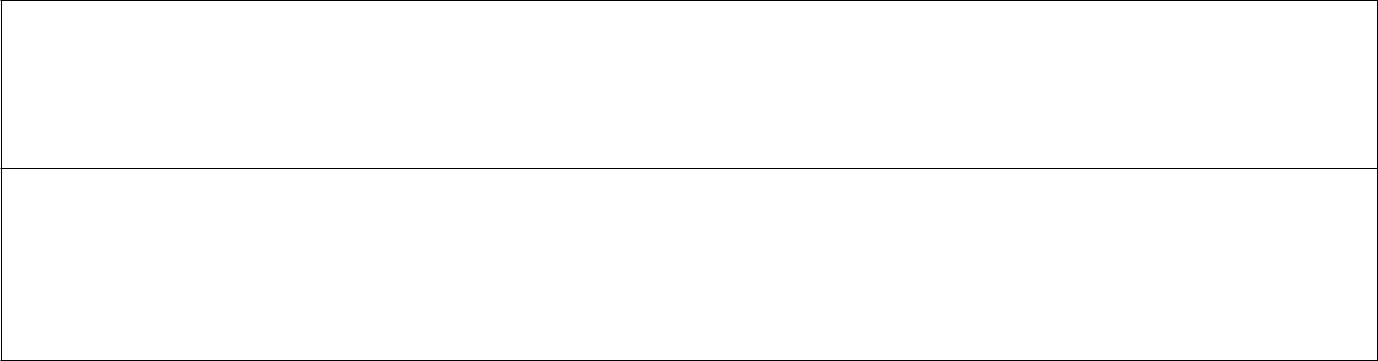
필요한 하드웨어 장치 및 이를 활용하는 방법 :

프로세서 내부에 하나의 플립/플롭을 두어 그 값으로 user mode와 커널 모드를 구분함

CPU가 실행하는 명령어의 주소 값에 따라, 그 명령어가 운영체제 명령어인지, 사용자 명령어인지를 구분하여 이 플립/ 플롭의 값을 제어함.

* Memory Protection이 필요한 이유는 무엇이며, 어떠한 방식으로 이루어지는가?

Memory Protection이 필요한 이유 :



두 개 이상의 프로세스가 메모리 공간을 공유(또는, 분할하여 사용)할 때, 하나의 프로세스가 다른 프로세스나 운영체제 의 메모리 공간에 접근하는 것을 방지하기 위함.

Memory Protection이 이루어지는 방법 :

프로세서 내부에 Base 및 Limit 레지스터를 두어 각각 해당 프로세스가 저장되어 있는 메모리 공간의 시작 주소와 그 크기를 저장함. CPU가 사용하는 명령어 및 데이터의 주소 값을 이 레지스터의 값들과 비교하여 범위를 벗어나는 경우, 프랩을 발생하여 프로세스 실행을 중단시킴.

* CPU Protection이 필요한 이유는 무엇이며, 어떠한 방식으로 이루어지는가?

CPU Protection이 필요한 이유 :

Multitasking 환경에서, 하나의 프로세스가 CPU를 너무 오래 독점하는 것을 방지하기 위하여 (IO 작업을 수행하지 않는 경우라도) 일정한 시간이 경과하면 프로세스 실행을 중단하고 다른 프로세스를 실행하기 위함.

CPU Protection이 이루어지는 방법 :

시스템 내부에 clock을 두어, 하나의 프로세스가 실행되기 시작한 이후 일정한 시간이 경과하면 인터럽트를 발생하여 (컨텍스트 스위치를 수행하여) 현재 프로세스의 실행을 중단하고 다른 프로세스의 실행을 시작함.

문제 2. <배점 : 5 + 10 + 5 = 20점> Interrupt

① 운영체제에 있어서, 인터럽트가 필요한 “이유”를 설명하시오.

* 1. 어떠한 사건(IO 종료, 특정한 사건의 발생, 프로그램 실행 오류 등)이 발생하였을 때, 이를 알림으로써 CPU가 Interrupt Service Routine을 실행하여 적절한 조치나 서비스를 받을 수 있도록 함.
  2. 특히, multiprogramming이나 multitasking 환경에 있어서는 IO를 CPU가 처리하지 않고 별도의 하드웨어 서브시스템에게 처리하도 록 하는 바, 하드웨어 서브시스템이 IO를 종료한 경우 이를 CPU에게 알림으로써 해당 프로세스의 실행을 계속할 수 있도록 함.
* 두 개의 프로세스 A, B가 있는 multiprogramming 환경(multitasking 환경이 아님)에서, 프로세스 A가 실행되는 중간에 하드디스 크로부터 512바이트의 데이터를 메모리로 읽어오고자 하는 IO가 요구되어 context switch 이후 프로세스 B가 실행된다고 가정 하자. 또한, 프로세스 B가 실행되는 도중에 IO가 완료되어 메모리에 이 데이터를 저장하는 과정을 가정한다. 이러한 일련의 과 정을 상세한 단계별로 구분하고 각 단계에서 수행되는 일을 간단하게 설명하시오, 단, 디바이스 컨트롤러, 인터럽트, 인터럽트 벡터, 인터럽트 서비스 루틴(ISR), DMA 등의 내용을 반드시 포함하시오.

∙ Execute process A

∙ I/O request by process A

∙ CPU loads registers in device controller for Hard Disk ∙ Execute process B after context switch

∙ While process B is running, Device controller moves 512bytes from disk to its local buffer ∙ While process B is running, Device controller triggers an interrupt

∙ CPU in User MODE is interrupted ∙ Context switch by OS

∙ Interrupt identification by OS

∙ CPU executes interrupt service routine to initiate DMA ∙ CPU loads registers in device controller for DMA

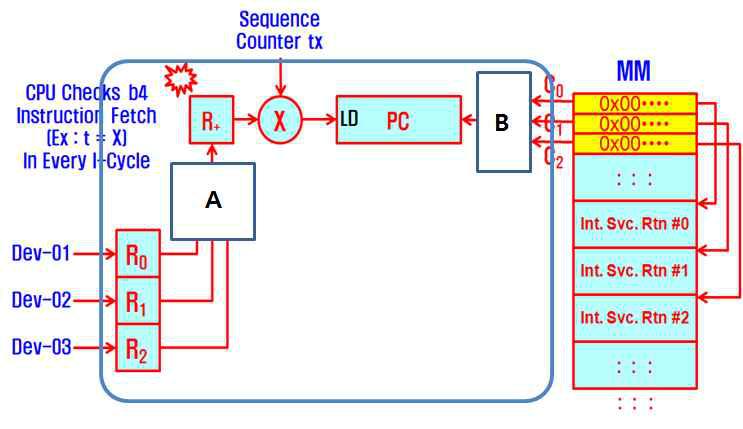
∙ CPU resumes interrupted process B (Since there are only processes A and B, and A is waiting for IO completion.) ∙ While process B is running, Device controller moves 512bytes from local buffer of disk to Memory using cycle stealing ∙ While process B is running, Device controller triggers an interrupt

∙ CPU in User MODE is interrupted ∙ Context switch by OS

∙ Interrupt identification by OS

∙ CPU executes interrupt service routine

∙ CPU resumes interrupted user process A or B

③ 우측의 그림은 인터럽트를 위한 하드웨어 개략도 이다. 빈 칸 A, B에 해당하는 “게이트”로 구성된 논리회로를 그리시오.

문제 1. <배점 : 5점 × 5 = 25점> Fundamentals of OS

* Multiprogramming의 개념이 도입된 이유를 설명하시오.
  + 한 프로그램이 IO를 실행하는 오랜 시간 동안 CPU의 활용도를 높이기 위하여, 다른 프로그램들을 실행함
* Multiprogramming 또는 multitasking 환경에서 운영체제가 필요한 이유를 설명하시오. 이 때, 컴퓨터의 핵심 자원인 CPU, 주

메모리, 하드디스크에 대하여 운영체제가 수행하는 역할들을 함께 설명하시오.

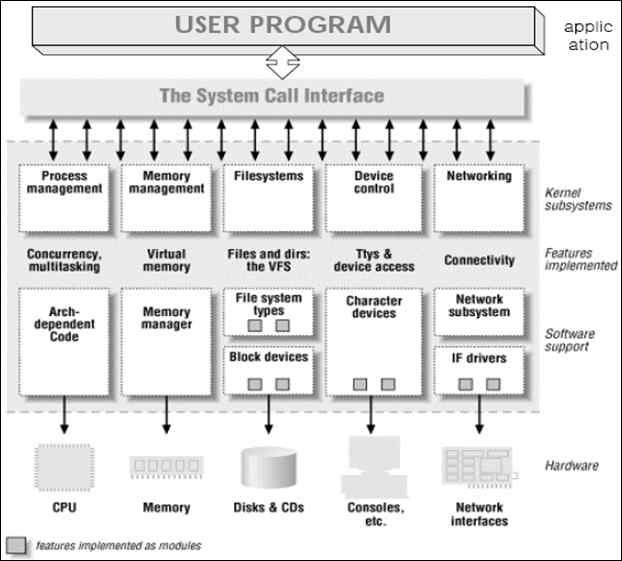
* + 필요한 이유 : 제한된 컴퓨팅 자원인 CPU, 주 메모리, 하드디스크 등을 2개 이상 다수의 프로그램이 **공유**함으로써 다양한 문제가 발생할 수 있으며, 이를 해결하기 위하여 시스템을 관리하는 소프트웨어가 필요함
  + CPU : 다수의 프로그램들이 순차적이고 효율적으로 CPU를 사용할 수 있도록 스케줄링이 필요함
  + 주 메모리 : 다수의 프로그램들이 제한된 메모리 공간을 분할하여 사용할 수 있도록 관리가 필요함
  + 하드 디스크 : 다수의 프로그램들이 제한된 영구저장 공간을 분할하여 사용할 수 있도록 관리가 필요함
* Multiprogramming 환경에 있어서, 한 프로그램이 IO operation을 최대한 배제하면 CPU를 오랜 시간 동안 독점할 수 있다. 이 와 같은 경우 발생하는 문제점을 적시하고 이를 해결하기 위한 방안을 제시하시오.
  + 문제점 : 해당 프로그램은 CPU를 최대한 활용하여 신속하게 실행될 수 있지만, 다수의 사용자나 다수의 프로그램 관점에서 본다면 waiting time (대기 시간)이 길어짐에 따라 시스템 전체의 효율이 낮아진다. 아울러, IO subsystem 또한 중요한 자원 (resource) 인데, 이들의 활용도가 낮아져서 시스템 전체의 활용도가 저하된다.
  + 해결 방안 : 이를 해결하기 위해서는, time sharing, 즉 multitasking 방식의 스케쥴링 알고리즘을 채택하는 것이 필요하다.
* Multiprogramming 또는 multitasking 환경에서, 프로그램이 실행되는 과정에서 IO의 시작 또는 종료, 특정 사건의 발생, 프로그

램 실행 오류 등이 발생할 수 있다. 이 때, CPU가 적절한 조치를 취하면서 일정한 스케줄링 알고리즘에 따라 프로그램들을 실행하기 위하여 필요한 메카니즘이 필요하다. 예를 들어, 임의의 프로그램(*Pa*)이 실행되는 중간에 위와 같은 이벤트가 발생한 경우 이를 처리하고 원래의 프로그램(*Pa*)으로 복귀하는 절차를 상세하게 설명하시오.

* + 인터럽트 처리 절차 :

⑤ 위 ④번 문제에 대한 “절차”에 있어서, context switch가 발생하는 시점을 모두 제시하시오.

- 1 -

문제 2. <배점 : 5점 × 3 = 15 점> System Call

① 우측에 주어진 그림에서, 운영체제에 포함되는 영역을 표시하시오.

② 우측의 그림에서 System call의 위치를 참조하여, System call이 필 요한 이유를 제시하시오.

커널 영역의 기능을 user mode가 사용 가능하게,

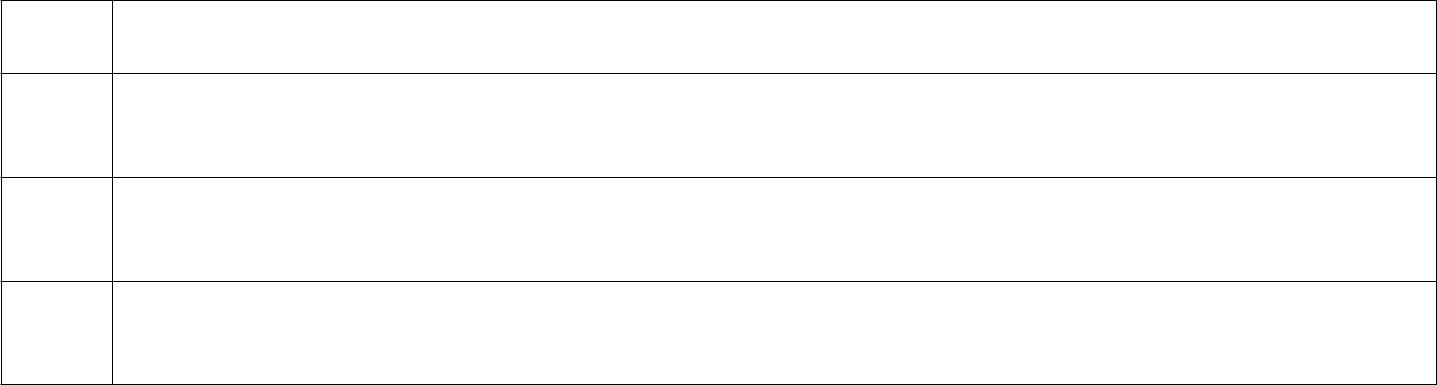
즉 process가 하드웨어에 직접 접근하여 필요한 기능을

사용할 수 있게 해준다.

③ 사용자 프로그램에 있는 fprintf() 함수가 실행되는 절차를 그림으로 그리고, 각 단계를 설명하시오.

문제 1. <배점 : 2 × 15 = 30점> Fundamentals of OS

* Device controller란 무엇이며, 왜 필요한가? 그리고 이 장치를 구성하는 요소에는 어떠한 것들이 포함되며, 이것이 수행하는 일은 무엇인가?



정의 자신이 제어하는 주변 장치와 자신의 로컬 버퍼 저장 장치 사이의 데이터 전송을 담당한다.

필요한 device들은 MM에 바로 접근할 수 없기 때문이다.

이유

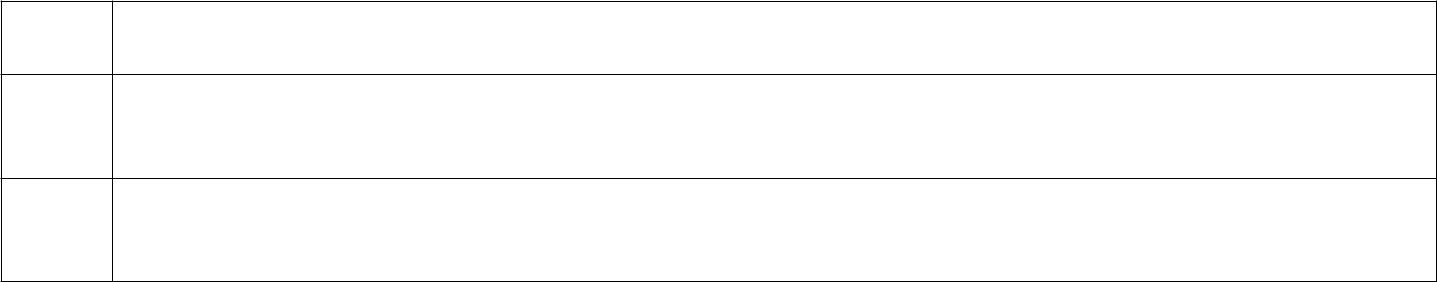
구성 로컬 버퍼 저장 장치, 특수 목적용 레지스터

요소

수행 자신의 레지스터의 내용을 조사하여 동작을 취하고, 장치로부터 자신의 로컬 버퍼로 데이터

업무 전송을 한다. 전송이 완료되면 인터럽트를 발생시킨다.

* Dual Mode란 어떠한 모드들로 구성되며 왜 필요한가? 그리고, 두 개의 모드를 구분하기 위하여 HW 또는 SW적으로 필요한 장치는 무엇이며, 이는 어떻게 활용되는가?



정의 운영체제가 두 개의 독립된 동작 모드, 사용자 모드(User Mode)와 커널 모드(Kernel Mode, Supervisor Mode, Privileged Mode)로 동작하는 것.

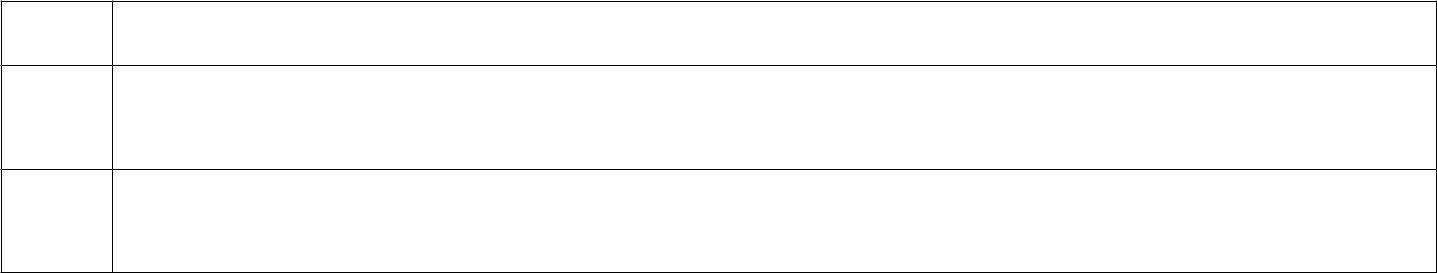
필요한 시스템 및 사용자를 보호하기 위하여

이유

필요한 (활용 방법 포함) CPU의 mode bit(flip-flop), 사용자 모드에서 시스템에 영향을 끼칠 수 있는 명령을 실행하려고

장치 시도하면 OS는 trap을 건다.

* DMA란 무엇이며, 왜 필요한가? 그리고, CPU를 사용하는 프로세스의 처리속도가 저하되는 이유는 무엇인가?



정의 Cycle Stealing

필요한

이유

속도저

하이유

dma(Direct Memory Acess)- 데이터블록 전체를 직접적으로 운반. Cycle Stealing은 DMA가 memory cycle을 CPU가 메모리 사 정의(CPU의 개입 없이 MM와 controller의 버퍼사이의 용하지 않을때만 사용하고 데이터 전송이 완성되면 DMA controller가 인터럽트를 발생시킨다.)-필요한이유(CPU는 Memory를 명령 수행하기 위해서 사용하는데 controller의 버퍼의 내용을 옮기는 것은 CPU에게 너무 오랜시간을 기다리게 하기때문)-속도저하이유(1.CPU가 memory 사용하지 않을 때 일어나지만 데이터를 buffer에서 MM으로 운반할 때 걸리는 시간이 발생 2.버스를 통해 전송되는 시간만큼 CPU가 메모리사용 불가능하기 때문에)

* Bootstrap 프로그램이란 무엇이며, 왜 필요한가? 그리고 이 프로그램이 수행하는 일은 무엇인가?



정의 컴퓨터가 구동을 시작하기 위한 초기 프로그램으로써, 시스템 초기화, 운영체제 로딩의 역할을 한다.

(필요한 이유) 시스템의 resource를 초기화하고, 운영체제의 커널을 메모리에 적재하기 위하여

필요한

이유 (ROM에 저장하는 이유) 항상 부팅하기 위해서는 Bootstrap 프로그램이 필요한데, ROM은 비휘발성 메모리로, 지워지지 않는 메모리이기 때문이다.

(운영체제 패키지에 포함시키지 않는 이유) Main Memory에는 처음에 운영체제가 적재되지 않기

때문에 Bootstrap 프로그램을 통해 MM에 OS를 적재하기 위함이다.

수행 시스템에 전원이 인가되면 Bootstrap 프로그램이 실행되며, 시스템의 resource를 초기화하고 운영체제의 kernel을 찾아 MM에 적재한다. 그 후 PC의 값을 MM의 OS 첫 번째 주소로 할당한다.

업무

① 두 개의 프로세스 A, B가 있는 multiprogramming 환경에서, 프로세스 A가 실행되는 중간에 하드디스크로부터 데이터를 메모리 로 읽어오고자 하는 IO가 발생하여, context switch 이후 프로세스 B가 실행된다고 가정하자. 또한, 프로세스 B가 실행되는 도 중에 IO가 완료되어 메모리에 이 데이터를 저장하는 과정을 가정한다. 이러한 일련의 과정을 간단하게 설명하시오, 단, **디바이**

**스** **컨트롤러**, **인터럽트**, **인터럽트 벡터**, **인터럽트 서비스 루틴**(ISR), DMA **등의 내용을 반드시 포함**하시오.



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 문제 1. <5점> 운영체제의 기능을 다섯 가지로 분류하고 각각을 간단하게 설명하시오. | | | | | | | |  |  |  |
|  |  |  |  |  |  |  | |  |  |  |
|  |  | 기능 |  |  |  | 설명 | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |
|  |  | 프로세스 관리 | |  | CPU에 프로세스와 스레드를 스케줄하기, 사용자 프로세스와 시스템 프로세스의 생성과 제거, 프로세스의 일시 중지와 재수행, 프로세스 동기화를 위한 기법 제공, 프로세스 통신을 위한 기법 제공 |  |  |  |  |  |

메모리의 어느 부분이 현재 사용되고 있으며 누구에 의해 사용되고 있는지를 추적해야 한다. 필요에 따라 메모리 공간을 할당하고 회수해야 한다.

메모리 관리

저장공간 관리

파일 생성과 제거, 파일을 보조 저장 장치로 매핑, 안정적인(비휘발성) 저장 매체에 파일 저장

(파일 시스템)

디바이스 관리

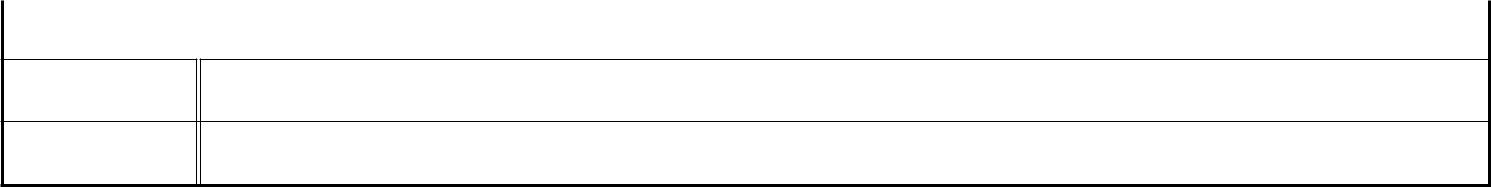
디바이스의 상태를 파악하여 관리하고, I/O의 스케줄링 및 전반적인 사항을 지시한다.

(드라이버)

네트워크

동기화와 프로세서의 스케줄링을 위한 프로세서 관리를 담당한다.

문제 2. <5점> 현재의 컴퓨터 시스템은 동시에 여러 개의 (ready) process를 허용하며, 이러한 (ready) process들은 하나의 메모리를 공유하며 저장되어 있다. 만일, 우리의 컴퓨터가 단지 하나의 프로세스만을 허용하며, 한다고 가정한다면, 운영체제의 기능 중 프로세스 관리 기능과 메모리 관리 기능은 어떻게 변화되어야 하는지 설명하시오.



|  |  |
| --- | --- |
| 기능 | 필요한 변화 |
|  |  |

프로세스 관리

메모리 관리

문제 3. <5점> Multiprogramming이란 무엇인가 간단히 설명하시오.

* + 한 프로그램이 I/O를 실행하는 오랜 시간 동안 CPU의 활용도를 높이기 위하여, 다른 프로그램들을 실행함

문제 4. <5점> Multiprogramming 환경에 있어서, 한 프로세스가 IO operation을 포함하지 않는다면 자신이 종료될 떄까지 CPU를 독점할 수 있다. 만일 이 프로세스가 실행되는 시간이 매우 길다면 다른 프로세스들이 기다려야 하는 시간이 너무 길어지는 문제가 발생한다. 이를 해결하기 위한 방안을 제시하시오.

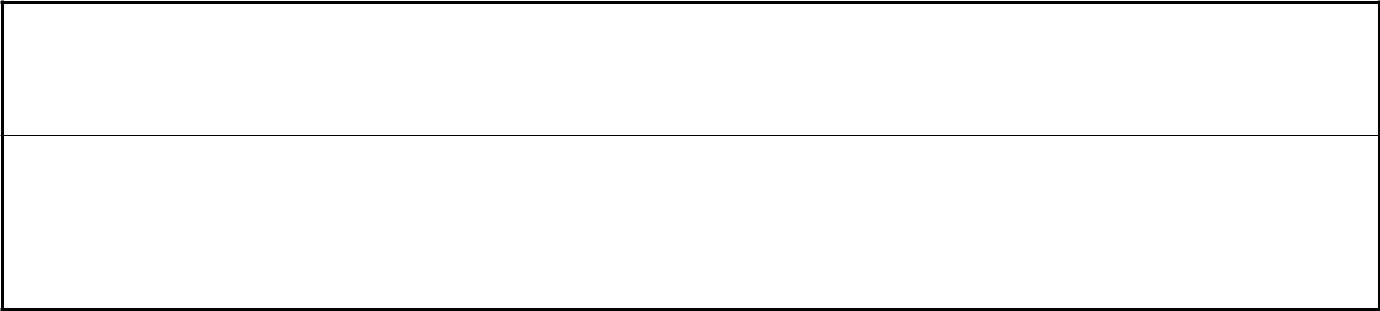
◦ 해결 방안의 명칭 (두 개의 명칭을 모두 쓰시오) : Time Sharing, Multitasking

◦ 해결 방안의 내용 : 정해진 시간 동안(time slice) 교대로 프로세스를 수행한다.

◦ 해결 방안에 필요한 하드웨어는 무엇인가? Clock

문제 6. <6점 × 3 = 18 점> 운영체제는 다양한 방법으로 protection 기능을 수행하고 있다, 다음의 질문에 답하시오.

* Dual Mode가 필요한 이유와 이를 위하여 필요한 하드웨어 장치 및 이를 활용하는 방법을 간단하게 기술하시오.

Dual Mode가 필요한 이유 :

시스템의 하드웨어(등 중요한 자원들을) 보호하기 위하여, 운영체제에게만 접근을 허용하고 사용자가 직접 접근하는 것 을 방지하기 위함.

필요한 하드웨어 장치 및 이를 활용하는 방법 :

프로세서 내부에 하나의 플립/플롭을 두어 그 값으로 user mode와 커널 모드를 구분함

CPU가 실행하는 명령어의 주소 값에 따라, 그 명령어가 운영체제 명령어인지, 사용자 명령어인지를 구분하여 이 플립/ 플롭의 값을 제어함.

* Memory Protection이 필요한 이유는 무엇이며, 어떠한 방식으로 이루어지는가? Memory Protection이 필요한 이유 :

두 개 이상의 프로세스가 메모리 공간을 공유(또는, 분할하여 사용)할 때, 하나의 프로세스가 다른 프로세스나 운영체제 의 메모리 공간에 접근하는 것을 방지하기 위함.

Memory Protection이 이루어지는 방법 :

프로세서 내부에 Base 및 Limit 레지스터를 두어 각각 해당 프로세스가 저장되어 있는 메모리 공간의 시작 주소와 그 크기를 저장함. CPU가 사용하는 명령어 및 데이터의 주소 값을 이 레지스터의 값들과 비교하여 범위를 벗어나는 경우, 프랩을 발생하여 프로세스 실행을 중단시킴.

* CPU Protection이 필요한 이유는 무엇이며, 어떠한 방식으로 이루어지는가?

CPU Protection이 필요한 이유 :

Multitasking 환경에서, 하나의 프로세스가 CPU를 너무 오래 독점하는 것을 방지하기 위하여 (IO 작업을 수행하지 않는 경우라도) 일정한 시간이 경과하면 프로세스 실행을 중단하고 다른 프로세스를 실행하기 위함.

CPU Protection이 이루어지는 방법 :

시스템 내부에 clock을 두어, 하나의 프로세스가 실행되기 시작한 이후 일정한 시간이 경과하면 인터럽트를 발생하여 (컨텍스트 스위치를 수행하여) 현재 프로세스의 실행을 중단하고 다른 프로세스의 실행을 시작함.