Operating System-Based Approaches to Cybersecurity

Architectural Support for OS in AI/ML/BigData/Cloud era. Essay

조민혁

Mobile System Engineering

32204292

목 차

1. 서론

1-1. 운영체제 개요

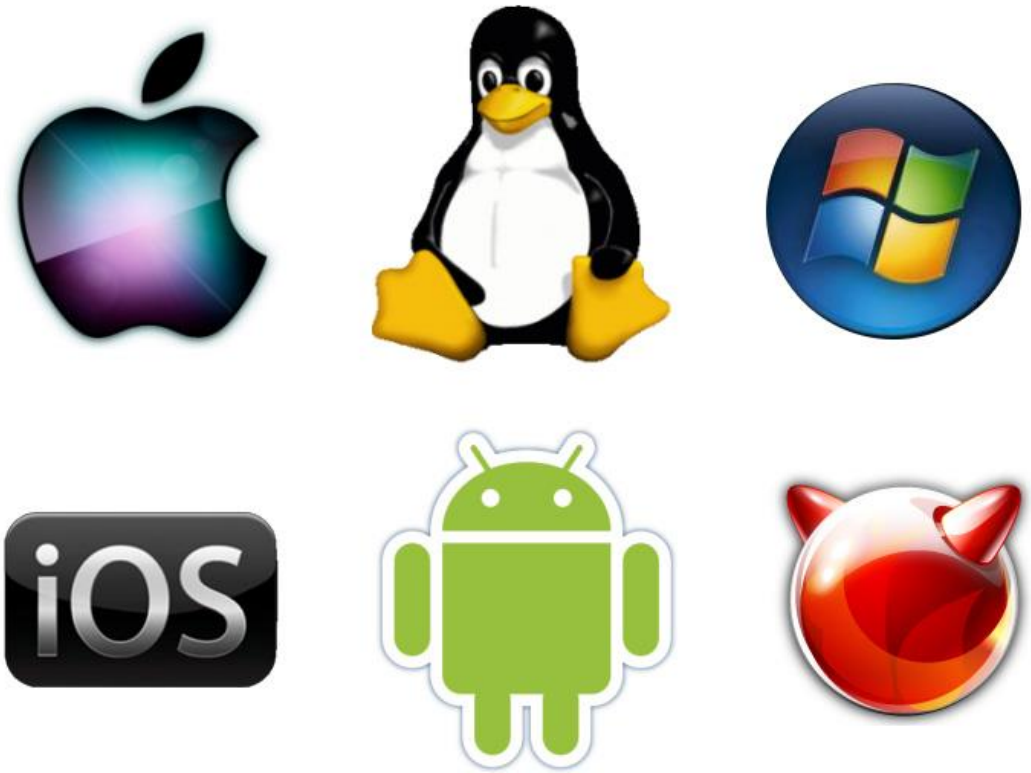

그림 1. 운영체제 종류

운영체제(Operating System)는 초창기 컴퓨터 시스템부터 현대의 첨단 컴퓨터 시스템에 이르기까지, 컴퓨터를 작동시키는 데 있어 필수적인 소프트웨어로 자리 잡았다. 운영체제는 컴퓨터 하드웨어(Hardware, HW)와 소프트웨어(Software, SW) 사이에 위치하며, 하드웨어 자원을 효율적으로 관리하고 소프트웨어가 하드웨어를 원활하게 사용할 수 있도록 돕는 중재자 역할을 한다. 대표적인 운영체제로는 마이크로소프트의 Windows, 애플의 macOS, 그리고 구글의 Android 가 있으며, 이들은 개인용 PC 와 스마트폰 시장에서 각각 높은 점유율을 유지하고 있다. 운영체제는 이러한 기기들에서 핵심적인 역할을 수행하며 필수적인 시스템 소프트웨어로 작동하고 있다.

운영체제는 하드웨어와 소프트웨어 사이의 복잡한 상호작용을 관리하는 핵심 요소로, 다음과 같은 방식으로 작동한다. 첫째, 운영체제는 하드웨어 자원의 추상화를 제공하여 소프트웨어가 하드웨어를 쉽게 사용할 수 있도록 한다. 이는 하드웨어의 복잡성을 숨기고 표준화된 인터페이스를

제공함으로써 개발자가 하드웨어 세부사항에 얽매이지 않고 소프트웨어를 개발할 수 있도록 한다. 둘째, 공유 자원에 대한 보호된 접근을 제공하여, 여러 프로그램이 동시에 실행되는 환경에서 자원 충돌을 방지한다. 이는 보안 및 인증 메커니즘을 통해 불법적인 접근을 방지하고 시스템의 무결성을 유지하는 데 기여한다. 셋째, 실행 중인 프로그램 간 통신과 협력을 지원하며, 이를 통해 프로그램 간 데이터 교환과 작업 조율이 가능하도록 한다.

운영체제는 또한 프로세스 관리와 자원 스케줄링을 통해 시스템 자원을 효율적으로 활용한다. 운영체제는 프로세스를 생성, 삭제, 스케줄링하며 실행 중인 프로세스 간의 우선순위를 결정한다. 이를 통해 CPU, 메모리, 디스크와 같은 자원을 최적화된 방식으로 배분하여 전체 시스템의 효율성을 극대화한다. 스케줄링은 특히 멀티태스킹 환경에서 중요한 역할을 하며, 다수의 작업이 동시에 실행될 때 각 작업이 필요한 자원을 적절히 할당받을 수 있도록 보장한다.

메모리 관리 역시 운영체제의 중요한 역할 중 하나이다. 운영체제는 가상 메모리(Virtual Memory) 기술을 통해 제한된 물리적 메모리 자원을 확장하여 사용한다. 가상 메모리는 디스크와 메모리 간의 협력을 통해 필요할 때마다 메모리 페이지를 디스크에서 메모리로 이동시키는 페이징(Paging) 기술을 사용한다. 이 과정은 프로그램이 실행에 필요한 메모리 공간을 동적으로 할당 받을 수 있도록 하며, 물리적 메모리의 크기 제한을 극복한다. 이러한 메커니즘은 현대 대규모 데이터 처리 및 고성능 컴퓨팅 환경에서 필수적인 기능으로 자리 잡고 있다.

운영체제는 파일 시스템 관리를 통해 데이터를 파일 형태로 저장하고, 파일 생성, 삭제, 읽기 및 쓰기 작업을 지원한다. 운영체제는 파일 접근 권한과 디렉터리 구조를 관리함으로써 데이터의 무결성과 보안을 보장한다. 또한, 파일 시스템의 효율적인 설계를 통해 데이터 검색 및 저장 속도를 최적화하며, 대규모 데이터 처리 환경에서도 안정적인 성능을 유지할 수 있도록 한다.

현대 사회에서 운영체제는 특정 기기와 사용 목적에 따라 특화된 형태로 발전하였다. 예를 들어, 스마트폰 운영체제는 사용자 친화적인 인터페이스와 배터리 최적화 기능에 중점을 두는 반면, 서버 운영체제는 대규모 데이터 처리와 안정성에 초점을 맞춘다. 최근 들어 AI/ML, 빅데이터, 클라우드 컴퓨팅과 같은 기술이 빠르게 발전함에 따라, 운영체제는 이러한 환경에서 더욱 중요한 역할을 담당하게 되었다. 예를 들어, AI 모델 학습을 지원하는 고성능 컴퓨팅 자원 관리, 빅데이터

처리에서의 분산 파일 시스템 최적화, 그리고 클라우드 환경에서의 가상화와 자원 격리를 포함한 다양한 기능이 요구된다. 이러한 내용은 1-2 절에서 더욱 구체적으로 다룰 예정이다.

본 에세이에서는 다음과 같은 주제를 다루고자 한다. 먼저, 서론에서 AI/ML/빅데이터/클라우드 시대에서의 운영체제의 역할을 살펴본다. 이후, 증가하는 사이버 위협과 운영체제 수준에서의 방어 필요성을 언급하여 보안의 필요성을 느낄 수 있도록 한다. 이후 본론에서 각 섹션별로 운영체제의 역할 및 작동 방식과 존재하는 위협들, 방어 전략을 살펴보도록 할 것이다. 이후 마지막 결론에서는 요약을 통해 내용을 정리하도록 하고, 현실적인 운영체제 보안의 한계를 언급하며 추가적인 대응 전략의 필요성을 강조한다. 마지막으로 미래 기술 발전에 따른 운영체제의 역할을 언급하고, 한 학기 동안의 운영체제 수업을 들으면서 느꼈던 생각들을 서술하며 에세이를 마무리한다.

1-2. AI/ML/빅데이터/클라우드 시대에서의 운영체제 역할

현대 기술에서 AI/ML, 빅데이터, 클라우드 컴퓨팅은 디지털 세계에서의 핵심적인 역할을 하고 있다. AI/ML 은 인공지능이 처음 언급된 1950 년 이후 빠른 속도로 발전해왔다. 1990 년대 통계학과 컴퓨터 과학이 융합된 결과물로 머신러닝이라는 개념이 등장했으며, 이는 데이터 기반 모델이 부각되었다. 2000 년대 이후 컴퓨터 성능이 급격하게 증가함에 따라 대규모 데이터셋이 축적되기 시작했다. 이에 따라 딥러닝 기술이 등장하였고, 이는 복잡한 패턴 인식, 음성 인식, 이미지 분류, 자연어 처리 등과 같은 여러 분야에서 뛰어난 성능을 보여주었다. 현대 AI/ML 은 인간과의 상호작용 중심 모델이 대두되고 있다. 너무나 유명한 GPT-모델, 초거대 언어 모델인 LLM 과 같은 혁신은 디지털 혁신을 가속화하고 있다.

빅데이터는 Volume, Velocity, Variety 라는 3V 특성을 기반으로 정의된다. 이는 방대한 데이터가 저장되고 분석되며 활용되는 현대 사회에서 중요한 특성이다. 현대 사회에서 존재하는 데이터의 양은 매우 많으며 현재에도 지속적으로 데이터가 축적되고 있다. 이에 따라 가공되지 않은 데이터를 가공된 형태로 데이터를 전처리 하는 것이 매우 중요하며, 최종적으로는 AI/ML 모델에 사용가능한 형태로 데이터를 만드는 것이 중요하다. 빅데이터는 인터넷 사용이 증가하며 급격하게 데이터 축적이 시작되었다. 데이터베이스와 같은 데이터 저장 기술이 존재했지만, 데이터의 증가 속도를 따라가기에는 한계가 있었다. 이에 따라 2000 년대 분산 시스템이 등장하여 병렬 처리 모델을

도입함으로써 대규모 데이터를 효과적으로 저장하고, 처리하였다. 현대 사회에서는 클라우드 컴퓨팅과 인공지능 융합을 통해 빅데이터 기술이 더욱 고도화 되었으며, 정형, 비정형 데이터를 통합 관리할 수 있는 환경을 제공하고, 클라우드 기술은 대규모 데이터를 유연하게 처리할 수 있도록 해주고 있다.

클라우드 컴퓨팅은 인터넷을 통해 데이터를 저장, 네트워크, 애플리케이션을 제공하는 서비스로 정의된다. IaaS, PaaS, SaaS 와 같은 서비스 모델을 통해 사용자 요구를 충족시킨다. 클라우드 컴퓨팅은 1960 년대 컴퓨터 자원을 공공 자원으로 제공할 수 있다는 개념에서 시작되었다. 이후, 가상화 기술이 발전하면서 하나의 물리적 하드웨어에서 여러 가상 머신이 실행될 수 있도록 하여 자원의 효율성을 극대화 하였다. 이후, 2000 년대 인터넷 기술이 발전함에 따라 아마존이 AWS 를 출시하여 클라루은 컴퓨팅 상용화를 시작하였다. AWS 를 통해 대규모 데이터 센터와 분산 시스템을 통해 유연하고 확장 가능한 클라우드 서비스를 제공할 수 있도록 하였다.

이러한 기술들의 발전은 운영체제 역할의 중요성을 더욱 강조해준다. 이에 따라 운영체제 설계 방식이 더욱 정교해지며, 고성능을 요구한다. 구체적으로는 대규모 데이터 처리, 파일 분산 시스템, 자원의 동적 관리를 요구한다. 이는 운영 체제가 단순한 자원 관리 소프트웨어 역할을 넘어서서 데이터 중심 시대의 핵심 요소로 진화하게 만들었다. 각 분야에 대해 구체적인 운영체제의 역할을 2 장에서 서술한다.

1-3. 증가하는 사이버 위협과 운영체제 수준에서의 방어 필요성

기술의 발전은 현대 사회에 빠른 속도의 변화와 편리함을 제공하며 다양한 영역에서 혁신을 이끌어왔다. 그러나 이러한 발전은 동시에 사이버 위협의 증가를 초래하며, 보안 문제를 더욱 심화시키고 있다. 디지털화가 가속화됨에 따라 개인, 기업, 공공기관이 의존하는 데이터와 시스템의 중요성이 커졌고, 이를 노리는 사이버 공격 역시 더욱 정교해지고 다양해졌다. 운영체제는 디지털 환경의 가장 기초적인 계층으로, 이러한 사이버 위협에 대응하는 데 있어 핵심적인 역할을 수행한다.

현대의 사이버 위협은 이전보다 훨씬 다양하고 심각한 양상을 보인다. 대표적으로, 랜섬웨어는 데이터를 암호화한 뒤 이를 복구하기 위해 금전을 요구하며, 기업과 공공기관에 막대한 경제적 피해를 초래한다. 실제로 Conti 와 같은 랜섬웨어는 2 년간 몇 백억을 갈취할 만큼 위협적인

랜섬웨어 공격이었다. 또한, 피싱 및 사회공학적 공격은 인간의 심리적 취약점을 악용하여 민감한 정보를 탈취하거나 시스템에 악성 코드를 설치한다. 제로데이 공격은 알려지지 않은 취약점을 악용해 보안 패치가 이루어지기 전에 시스템에 접근하며, 이는 운영체제와 같은 핵심 소프트웨어를 직접적으로 위협한다. 클라우드 환경의 멀티테넌트 구조에서 발생하는 데이터 유출 및 가상 머신 탈출 공격도 현대 디지털 환경에서 큰 우려로 대두되고 있다.

이러한 사이버 위협은 운영체제 설계와 구현 방식에 새로운 도전 과제를 제시한다. 운영체제는 데이터 보호와 시스템 자원의 무결성을 유지하기 위해 중요한 보안 기능을 제공해야 한다. 첫째, 운영체제는 접근 제어 및 사용자 인증을 통해 불법적인 시스템 접근을 차단한다. 둘째, 운영체제는 실시간 위협 탐지와 방어 메커니즘을 갖추어야 한다. 머신러닝 기반 위협 탐지 시스템은 비정상적인 패턴을 실시간으로 분석하고 악성 코드를 차단하는 데 효과적으로 활용될 수 있다. 셋째, 운영체제는 데이터 암호화와 무결성 검증 기술을 통해 데이터 유출과 변조를 방지한다. 이를 통해 데이터는 저장 및 전송 과정에서 보호받을 수 있다.

특히, 클라우드와 같은 가상화 환경에서는 운영체제가 자원의 격리와 가상화 기반 보안을 통해 멀티테넌트 환경에서의 공격을 방지해야 한다. 하이퍼바이저 보안과 컨테이너 샌드박싱은 가상 머신과 컨테이너 간의 상호 간섭을 방지하여 각 사용자의 데이터를 안전하게 보호한다. 더 나아가, 운영체제는 자원의 동적 관리를 통해 시스템 성능을 유지하며, 동시에 사이버 위협에 대비하는 방어 메커니즘을 제공해야 한다.

결론적으로, 기술 발전은 사이버 위협의 정교화와 증가를 불가피하게 동반하고 있다. 이러한 환경에서 운영체제는 디지털 환경의 최전방 방어선으로서, 시스템의 안정성과 데이터를 보호하는 역할을 수행한다. 운영체제가 제공하는 보안 메커니즘은 사이버 위협을 예방하고 대응하는 데 필수적이며, 미래의 운영체제 설계는 이러한 보안 기능을 더욱 강화하는 방향으로 나아가야 한다. 이에 따라, 본 에세이에서는 본론에서 존재하는 사이버 위협들과 대응할 수 있는 방어 전략을 살펴보고자 한다.

2. 본론

2-1. AI/ML

2-1-1. AI/ML 환경에서의 운영체제

1 장에서 언급했듯 AI/ML 은 현대 디지털 기술에서 가장 혁신적인 기술로 자리잡고 있다. 이에 따라 본 절에서는 AI/ML 환경에서의 운영체제 역할을 구체화하려 한다. AI/ML 환경은 대규모 데이터 처리, 고성능 연산, 그리고 복잡한 모델 학습 과정을 포함한다. 이에 따라 운영체제는 자원 관리와 최적화를 중심으로 AI/ML 환경에서의 운영체제 역할을 해야 한다. AI/ML 작업은 언급했듯이 대규모 연산은 수반한다. 이는 고성능 GPU 와 TPU 와 같은 특수 하드웨어를 필요로 하며 운영체제는 이러한 하드웨어 자원을 효율적으로 관리하고 스케줄링 하는데 핵심적인 역할을 한다. AI/ML 작업은 병렬 처리 환경에서 다수의 AI 작업이 실행될 때, 운영체제는 작업 간 자원 할당을 최적화하여 연산 성능을 극대화해야 한다. 예를 들어, GPU 스케줄링은 AI 모델의 학습 속도와 효율성을 결정 짓는 중요한 요소로, 운영체제는 GPU 활용도를 최대화하면서 작업 간 충돌을 방지하는 역할을 수행한다.

또한, AI/ML 모델 학습은 방대한 양의 데이터를 필요로 한다. 이는 데이터 입출력(I/O) 성능이 학습에 중요한 영향을 미친다. 운영체제는 데이터 입출력 속도를 최적화하기 위해 캐싱과 프리패칭 기술을 활용하여 데이터를 효율적으로 관리할 수 있다. 특히, 가상 메모리 기술을 극대화하여 대규모 데이터셋이 디스크에 저장된 경우, 운영체제는 메모리와 디스크 간 데이터 전송을 최적화하여 병목 현상을 줄이도록 한다.

가상 메모리 기술의 사용을 구체적으로 기술하면 AI/ML 은 제한된 물리적 메모리를 초과하는 대규모 메모리 공간을 요구한다. 이에 따라 운영체제는 페이지 교체 알고리즘과 같은 메커니즘을 통해 자주 사용되는 데이터를 메모리에 유지하고, 불필요한 데이터를 디스크에 옮기는 작업을 최적화할 수 있다. 이에 따라 AI/ML 작업의 연속성을 보장 가능하다.

2-1-2. AI/ML 환경에서의 존재하는 위협들

해당 AI/ML 기술과 운영체제의 협력은 매우 효율적이다. 그러나 대규모 데이터 셋과 AI 모델 중요한 지적 재산권과 관련된 중요한 자산으로 간주되며 경쟁력의 핵심으로 평가된다. 이에 따라 이에 대한 불법적인 접근과 도난을 방지해야 한다. 이에 따라 이러한 사이버 위협들이 몇 가지 존재한다.

먼저, 데이터 유출이다. AI/ML 모델 학습에 사용되는 데이터셋은 민감한 개인 정보나 기밀 데이터를 포함할 수 있다. 데이터 유출은 공격자가 중요 정보를 탈취하고, 불법적으로 활용하는 상황을 초래할 수 있다. 예를 들어 환자의 의료 데이터를 학습한 AI 모델의 경우 환자의 개인 정보가 포함된 민감한 데이터가 유출된다면 심각한 윤리적 법적 문제가 발생할 수 있다.

둘째, 모델 도난 및 리버싱이다. AI/ML 모델은 오랜 시간과 비용을 투자해 만든 중요한 자산으로 존재한다. 이에 따라 모델이 도난되거나, 역공학을 통한 모델 복제, 모델의 출력을 분석하여 내부 구조의 재구성은 심각한 보안 문제로 이어진다.

셋째, 데이터 중독 공격으로 학습 데이터셋에 악의적인 데이터를 의도적으로 삽입하여 모델의 학습 과정을 방해하여 모델의 성능을 저하시킬 수 있다. 또한 악의적인 목적으로 AI/ML 모델을 사용한다면 문제가 될 수 있다. AI/ML 모델이 악의적인 목적으로 재구성된다면 LLM 과 같은 모델에서는 허위 사실 확산, 피싱 이메일 생성에 악용될 수 있고, 이미지 생성 모델은 딥페이크와 같은 범죄 행위로 악용될 수 있다.

마지막으로 사이드 채널 공격이 존재하는데, 이는 모델 도난 및 리버싱과 유사한 공격으로 공격자는 하드웨어의 전력 소비, 처리 시간, 전자파 방출 등을 분석해 모델 내부 구조나 학습 데이터를 추론할 수 있다. 따라서 수많은 사이버 위협이 존재하므로 1 차적으로 운영체제 차원에서 보호가 되어야한다. 이러한 보안 전략은 2-1-3 절에서 자세히 서술한다.

2-1-3. AI/ML 환경에서의 운영체제 차원에서의 보호 및 방지

앞선 절에서 언급했듯 해당 절에서는 논의된 사이버 위협들에 대한 운영체제 차원에서의 보호 및 방지를 알아보도록 한다. 운영체제는 다양한 차원에서의 보안 메커니즘을 설계하고 구현해야 할 필요가 있다. 개략적으로 운영체제는 AI/ML 환경에서 데이터 보호, 자원 격리, 실시간 위협 탐지, 모델 무결성 유지와 같은 보안 기능을 제공함으로써 사이버 공격으로부터 시스템을 보호할 수 있다. 이를 구체적으로 서술하면 다음과 같다.

먼저 데이터 보호를 위해 운영체제 차원에서는 데이터를 저장하거나 전송할 때 암호화 기술을 사용할 필요가 있다. 이에 따라 저장된 데이터는 AES, RSA 와 같은 암호화 알고리즘을 사용해야 하며, 전송중인 데이터는 TLS 프로토콜을 사용해 네트워크 상의 데이터 유출을 방지해야 한다. 데이터를 주고받을 때는 해시 기반 알고리즘을 활용하여 무결성 검증을 철저히 하여 데이터 보호를 제공해야 한다.

다음, 모델 도난 및 변조 방지를 위해서는 운영체제는 인증된 사용자만이 모델을 실행하거나 수정하도록 해야 한다. 즉, 모델 파일에 디지털 서명을 적용하여 무결성을 검증해, 모델 변조를 방지하도록 한다. 또한, 운영체제는 모델 실행 중 비정상적인 동작을 탐지하고, 스케줄링 과정에서 의심스러운 프로세스를 탐지하고 차단하는 조치를 취할 수 있어야 한다.

셋째, 데이터 중독 공격 방지를 위한 운영체제 기능은 다음과 같다. 운영체제는 데이터 소스의 신뢰성을 인증하고, 신뢰할 수 없는 외부 데이터 소스에서의 데이터 다운로드를 사전에 차단하는 기능을 제공해야 한다. 또한, 데이터 셋을 사용하기 전에 데이터 검증 및 정제 과정을 지원함으로써 데이터 무결성을 확인하고 신뢰할 수 있는 데이터를 AI/ML 모델에 사용하도록 한다.

마지막으로 사이드 채널 공격 방지에 대한 운영체제의 보안 지원은 다음과 같다. 운영체제는 가상화 기술을 사용하여 하이퍼바이저 기반의 VM 격리를 하도록 하여 사용자가 다른 사용자 자원에 접근할 수 없도록 한다. 하이퍼바이저 기반의 VM 격리 기술은 각 하드웨어, 메모리, CPU, 네트워크, 디스크와 같은 하드웨어 자원을 사용자마다 나누어 VM 마다 할당하여 격리하는 기술이다.

또한 정보 누출 방지를 위해서 하드웨어 지원 보안 기술인 Intel SGX, AMD SEV 와의 통합을 통해 보안을 강화해야 한다. Intel SGX 는 응용 프로그램의 일부 데이터를 암호화된 메모리 영역에

격리하여 실행하는 하드웨어 기반 보안 기술로써 민감한 데이터를 CPU 외부, 운영체제, 하이퍼바이저, 관리자 계정도 접근할 수 없게 하는 것이다. AMD SEV 는 전체 가상 머신의 메모리를 암호화하여 보호하는 기술로 공격자로부터 보호한다. 이를 통해 각 VM 마다 암호화 키를 할당하여, 다른 VM 이 다른 VM 의 메모리를 읽거나 변경할 수 없다. 따라서 이러한 기술을 통합하여 보안을 강화한다면 운영체제 차원에서 1 차적으로 보호할 수 있을 것이다.

2-2. 빅데이터

2-2-1. 빅데이터 환경에서의 운영체제

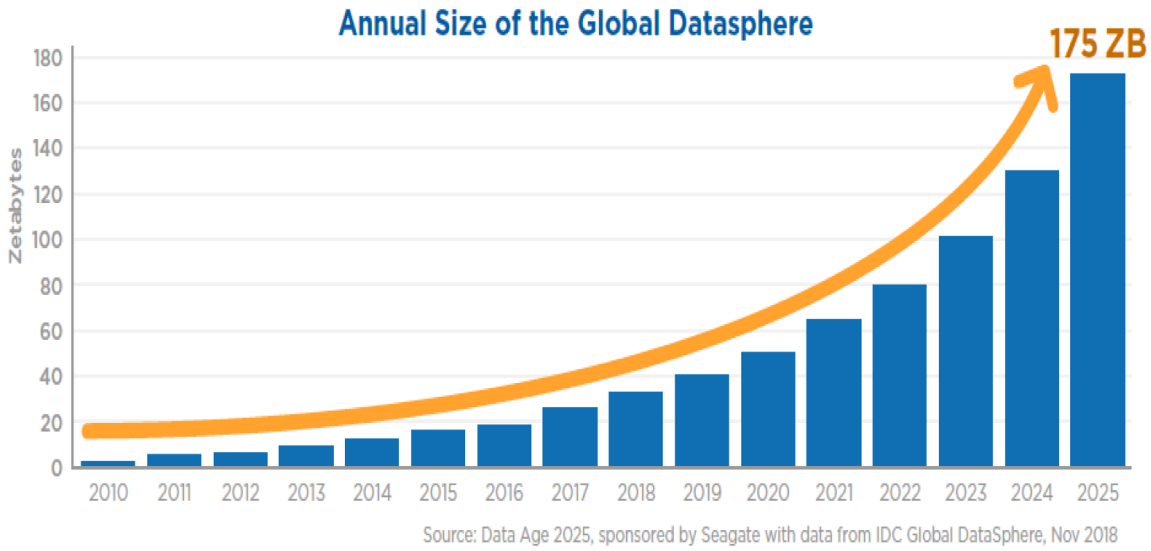

그림 2. 인터넷 상 데이터 양 예측 그래프

빅데이터 처리는 대규모 데이터를 기존 시스템이 처리할 수 있는 한계를 극복하여 다룰 것을 요구한다. 특히, 그림 2 와 같이 2025 년 에는 175ZB 의 데이터가 축적될 것이라 예상하고 있으며 이 중 90%는 가공되지 않은 데이터로 특정되고 있다. 이에 따라 운영체제는 방대한 양의 데이터를 효율적으로 관리하고 처리하기 위해 중요한 역할을 수행한다.

먼저, 운영체제는 분산 파일 시스템 관리를 하도록 지원해야 한다. 빅데이터는 하나의 물리적 시스템에 저장되기에는 데이터 크기가 너무 크기 때문에, 여러 노드에 데이터를 분산 저장하는 분산 파일 시스템이 필요하다. 대표적인 분산 파일 시스템은 GFS, HDFS 가 있다.

또한 자원 관리와 작업 스케줄링을 제공해야 한다. 특히, 빅데이터 처리 작업은 병렬 처리를 통해 수행되기 때문에, 여러 컴퓨팅 노드의 자원을 효과적으로 관리할 것을 요구한다. 이에 따라 작업 우선순위와 데이터 로컬리티를 고려한 작업 스케줄링을 수행해야 한다. 또한 병렬로 처리되기에 openMP, MPI, pthread 와 같은 병렬 처리 기법을 수행하며 적절한 동기화 기법을 사용하여 CPU 및 메모리 자원 간에 충돌을 방지해야 한다.

마지막으로 대규모 데이터에 대한 빠른 디스크 입출력 성능을 요구할 것이기에 운영체제는 데이터 접근에 대한 최적화가 필요하다. 이를 위해 운영체제는 캐싱을 통해 자주 사용되는 데이터에 대한 캐시를 진행해 데이터 접근 속도를 높인다. 또한 예상되는 데이터 접근 패턴을 기반으로 데이터를 미리 로드해 입출력 속도를 높이도록 한다. 그리고, 다수의 입출력 작업을 병렬로 처리하여 디스크 접근 시간을 단축하도록 한다.

이러한 운영체제의 설계와 구현을 통해 빅데이터에 대한 적절한 처리가 이루어지고 안정적인 빅데이터 작업이 이루어질 수 있다.

2-2-2. 빅데이터 환경에서의 존재하는 위협들

데이터의 양이 기하급수적으로 증가하는 현대 컴퓨터 시스템에서의 운영체제를 통한 빅데이터 처리 작업을 매우 효율적이다. 그러나, 가공되지 않은 데이터와 가공된 데이터 모두 사이버 위협이 존재할 수 있다. 특히, 민감한 정보에 대한 데이터는 위협당하지 않도록 더욱 주의를 가해야 한다. 빅데이터 환경에서 존재하는 위협들은 다음과 같다.

먼저, AI/ML 에서의 취약점과 유사하게 데이터 유출 및 권한 없는 접근에 대한 사이버 위협이 존재한다. 빅데이터 환경에서의 데이터는 언급했듯 개인 정보 및 금융 정보와 같은 민감한 정보를 포함한다. 이에 따라 데이터가 네트워크를 통해 전송될 때 네트워크 스니핑을 통해 패킷이 가로채져 공격에 노출될 수 있으며, 데이터가 저장될 때 권한 설정이 적절하지 못할 시 허가 받지 않은 사용자가 데이터에 접근하여 공격에 노출될 수 있다.

다음, 분산 환경에서의 노트 취약점 및 내부 환경이 존재한다. 빅데이터 환경은 여러 컴퓨팅 노드에 데이터를 분산하여 처리한다고 2-2-1 절에서 서술하였다. 이에 따라 하나의 노드에서 발생하는 보안 취약점이 전체 시스템에 영향을 미칠 수 있다. 만약 공격자가 하나의 노드에서 시스템 접근 권한을

획득하는 권한 상승 공격을 일으킨다면 데이터를 탈취하고 전체 시스템이 마비될 수 있는 문제점을 야기한다. 또한 공격자가 신뢰할 수 없는 노드에서 악성 코드를 실행하여 데이터 손상 및 시스템 장악을 유발할 수 있다.

마지막으로 빅데이터 시스템에 대한 DDos 공격으로 인해 빅데이터 클러스터의 서비스 중단이 발생할 수 있다. 실제로 빅데이터 시스템은 네트워크를 통해 데이터를 주고받는 환경이 존재하기에, 만약 공격자로부터 악의적인 수많은 트래픽을 차단할 수 없는 환경이라면 이는 DDos 환경에 노출되어 시스템의 마비를 야기할 수 있다.

2-2-3. 빅데이터 환경에서의 운영체제 차원에서의 보호 및 방지

빅데이터 환경에서 발생하는 다양한 사이버 위협과 보안 취약점에 대응하기 위해 운영체제 차원에서 적절한 보호 및 공격 방지 메커니즘을 제공해야 한다. 이에 따라 본 절에서는 앞서 언급된 위협들에 대응하기 위해 운영체제 차원의 보호 및 방지 전략을 논의한다.

먼저, 운영체제는 데이터를 저장하고 전송하는 과정에서 암호화 기술을 적용해야 하는데 이는 2-2 절에서 언급했기에 생략한다. 또한, 데이터 접근 권한을 운영체제 차원에서 세분화하고, 역할 기반 접근 제어를 통해 민감한 데이터에 대한 불법 접근을 방지해야 한다. 또한, 다단계 인증을 통해 운영체제는 사용자 인증 과정을 강화해야 하며, 인증되지 않은 접근을 차단할 수 있도록 해야 한다.

두 번째로는, 분산 시스템의 노드를 보호할 수 있도록 해야 한다. 운영체제는 각 노드에 대한 신뢰할 수 있는 노드 인증 메커니즘을 구축할 수 있어야 한다. 예를 들어 X.509 인증서를 활용해 노드 인증 기술을 도입하여 효과적인 노드 보호가 이루어져야 한다. 또한, 운영체제는 자체적으로 악성 노드 탐지 시스템을 구현하여 비정상적인 동작을 보이는 노드를 탐지하고, 그러한 노드를 격리할 수 있는 적절한 기술을 구축해야 한다. 이런 기술은 머신러닝 기반의 이상 탐지 알고리즘을 활용해 악성 코드 실행 및 비정상적인 트래픽 패턴을 실시간으로 모니터링 할 수 있게 할 수 있다.

마지막으로 네트워크 차원에서 공격을 방지하기 위해 운영체제는 소프트웨어 정의 네트워킹을 활용하여 공격 트래픽을 자동으로 우회하거나 차단하도록 해야 한다. 또한 운영체제 차원에서 IP 단위 또는 사용자 세션 단위로 요청 횟수를 제한하여 서버 과부하를 방지할 수 있도록 해야 한다. 그리고, 트래픽 흐름에 대한 속도를 제한하여 DDos 로 인한 서비스 중단을 방지하도록 해야 한다.

운영체제는 빅데이터 환경에서 발생할 수 있는 데이터 유출, DDoS 공격, 노드 취약점 등 다양한 위협에 대응하기 위한 핵심 방어선이다. 데이터 암호화, 분산 노드 보호, DDoS 방어 메커니즘, 그리고 데이터 무결성 검증과 같은 전략은 빅데이터 시스템의 안전성과 신뢰성을 보장하는 데 필수적이다. 향후 빅데이터 시스템의 확장과 발전에 따라 운영체제는 더욱 정교하고 강력한 보안 기능을 지속적으로 제공해야 한다.

2-3. 클라우드 컴퓨팅

2-3-1. 클라우드 환경에서의 운영체제

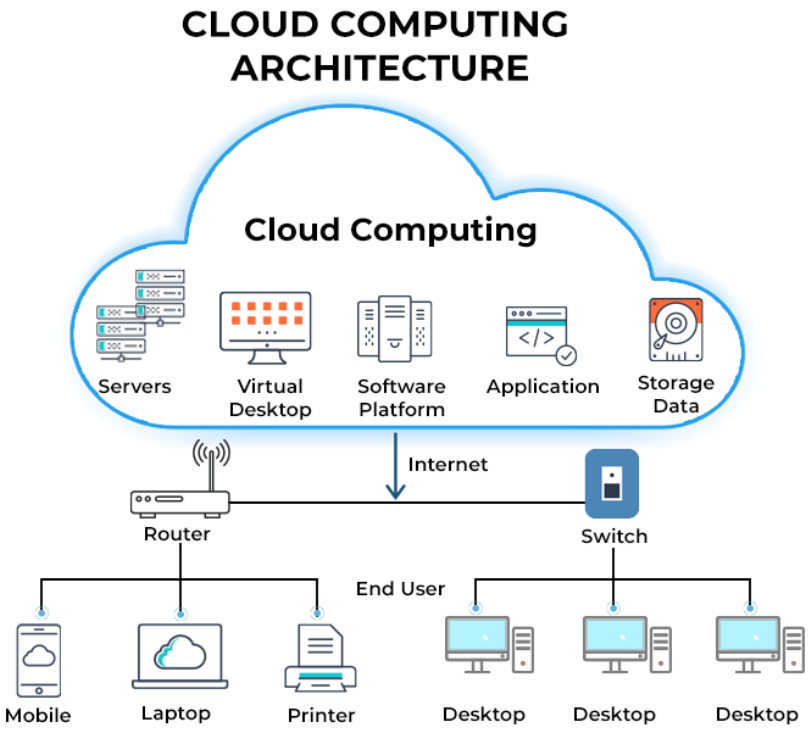

그림 3. 클라우드 컴퓨팅 도식화

앞서 언급했듯 클라우드 컴퓨팅은 인터넷을 통해 데이터를 저장, 처리, 그리고 관리하는 서비스이며 다양한 서비스 모델에서 운영체제는 핵심적인 역할을 한다. 먼저, 운영체제는 가상화 기술을 제공하여 물리적 자원을 논리적으로 분리하여 여러 가상 시스템에서 효율적으로 활용할 수 있게 한다. 하이퍼바이저를 통해 물리적 하드웨어를 여러 가상 머신으로 분리하여, 클라우드 환경에서 VM 이 독립적으로 실행될 수 있도록 한다. 그리고, 도커와 쿠버네티스와 같은 컨테이너

기술을 지원하여 애플리케이션의 독립성과 이동성을 제공한다. 컨테이너는 운영체제 수준에서 격리되어 실행되는데, 애플리케이션과 실행에 필요한 모든 요소를 하나의 패키지로 묶어 독립적인 실행 환경을 제공할 수 있도록 한다.

운영체제는 자원 관리와 최적화를 제공해준다. 클라우드 환경에서는 다수의 사용자가 동일한 물리적 자원을 공유하므로 운영체제는 사용자의 요구에 따라 CPU, 메모리, 네트워크 대역폭 등을 동적으로 할당한다. 또한 로드 밸런싱을 통해 운영체제는 여러 가상 머신 또는 컨테이너에 작업을 분산시켜 리소스 과부하를 방지한다. 그리고, 통신에 필요한 네트워크 자원을 동적으로 구성하고 관리하여 자원 관리와 최적화를 제공할 수 있도록 해준다.

마지막으로, 운영체제는 대규모의 데이터와 사용자를 처리하기 위해 높은 확장성과 가용성을 제공해준다. 오토스케일링을 통해 클라우드 워크로드에 따라 자원을 자동으로 확장하거나 축소하여 비용을 최적화하고 성능을 보장해준다. 그리고, 운영체제는 데이터를 여러 물리적 서버에 데이터를 복제하여 장애 발생 시 데이터 손실을 방지하도록 한다. 예를 들어 Amazon S3 와 같은 클라우드 스토리지 서비스는 운영체제 수준에서 데이터 복제 및 복구를 지원하도록 한다.

클라우드 환경에서 운영체제는 가상화 기술, 자원 관리, 보안 강화, 확장성과 같은 다양한 기능을 수행하며 클라우드 서비스의 효율성과 신뢰성을 보장한다. 이러한 운영체제는 사용자의 요구에 따라 자원을 동적으로 조정하고, 서비스의 연속성을 보장하기 위해 지속적으로 발전하고 있다. 클라우드 기술의 발전과 함께 운영체제는 더욱 정교하고 강력한 역할을 수행하게 될 것이다.

2-3-2. 클라우드 환경에서의 존재하는 위협들

클라우드 컴퓨팅 기술이 발전할수록 사이버 위협은 그에 따라 발전해왔다. 이번 절에서는 클라우드 환경에서 발생하는 사이버 위협들을 알아보도록 한다. 먼저, 컨테이너 탈출 공격이 발생할 수 있다. 컨테이너 탈출 공격은 컨테이너 내에서 실행되는 애플리케이션이나 프로세스가 컨테이너의 격리된 황경에서 벗어나 호스트 시스템이나 다른 컨테이너에 접근하는 공격이다. 공격자는 컨테이너 런타임이나 커널 취약점을 악용하여 호스트 운영체제에 접근해 더 많은 권한을 얻은 후 호스트 시스템의 파일 시스템, 네트워크, 메모리 등을 공격할 수 있다. 또한 동일한 호스트에서 실행 중인 다른 컨테이너에 접근해 데이터를 탈취하여 악용이 가능하다.

두 번 째로 컨테이너 공급망 공격이 가능하다. 공격자는 컨테이너 이미지의 빌드 과정이나 배포 경로를 악용하여 악성 코드나 백도어를 주입 가능하다. 공격자가 컨테이너 이미지를 생성하는 과정에서 빌드 프로세스에 침투하거나 악성 코드를 삽입한 후 컨테이너 이미지를 저장하는 레지스트리를 타겟으로 공격이 가능하다. 배포 단계에서 네트워크 트래픽을 가로채어 악성 이미지로 교체하고, 실행 중인 컨테이너 이미지를 타겟으로 공격하여 악성 명령을 실행하도록 할 수 있다. 이에 따라 컨테이너 이미지가 공급되는 모든 과정에서 공격이 가능해지며 이미지가 적절히 공급되도록 보장해주는 것이 중요하다.

마지막으로 컨테이너 자원 고갈 위협이 존재한다. 공격자가 컨테이너 내에서 무한 루프, 과도한 네트워크 요청, 메모리 소비를 유발하여 시스템 자원을 고갈시키고 서비스를 중단시키는 행위이다. 이를 통해 컨테이너가 자원을 독점하여 다른 서비스가 실행되지 못하도록 방해하고, 호스트 시스템 전체가 다운되도록 하는 위협이다. 구체적으로는 ‘’WHILE :; DO :; DONE :;” 과 같은 명령어를 통해 CPU 자원을 독점할 수 있고, 대량의 메모리를 할당하고, 메모리 누수를 유발하는 애플리케이션을 실행하여 Out of Memory 킬러를 활성화 시켜 다른 컨테이너가 강제로 종료되도록 할 수 있다. 그리고, 디스크 쓰기 작업 반복 및 대량의 임시 파일 생성을 통해 디스크 공간을 고갈시킬 수도 있다. 이를 통해 디스크 I/O 병목 현상이 발생해 성능 저하를 초래할 수 있는 위협이 존재한다.

2-3-3. 클라우드 환경에서의 운영체제 차원에서의 보호 및 방지

클라우드 환경에서의 사이버 위협은 점점 더 복잡해지고 다양해진다. 운영체제는 이러한 위협들을 방지하기 위해 중요한 역할을 수행해야 하며, 크게는 컨테이너 격리, 이미지 무결성 보장, 자원 관리 등을 통해 보안성을 유지해야 한다. 본 절에서는 앞서 언급된 사이버 위협들을 대상으로 운영체제 차원에서의 보호를 알아보도록 한다.

먼저, 운영체제는 컨테이너 탈출 공격을 방지하기 위해 적절한 메커니즘을 제공해야 한다. 네임스페이스를 활용하여 컨테이너의 프로세스, 파일 시스템, 네트워크를 독립적으로 격리하도록 해줘야 한다. 또한, Seccomp 을 통해 컨테이너에서 실행 가능한 시스템 호출을 제한하도록 한다. 그리고, AppArmor 또는 SELinux 를 통해 컨테이너의 파일 및 프로세스 접근 권한에 대해 세부적으로 설정하여 격리 수준을 강화할 수 있도록 운영체제는 제공해야 한다.

두 번째로 운영체제는 컨테이너 공급망 공격을 방지하기 위해, CI/CD 파이프라인 보안을 강화하는데, 빌드 스크립트와 레지스트리 접근 권한을 최소화하여 공격자가 침투할 수 있는 경로를 최대한 차단하도록 해야 한다. 또한, 운영체제는 컨테이너 이미지가 저장되는 레지스트리에 접근할 수 있는 계정을 제한하고, 다단계 인증 메커니즘을 제공하여 사용자 인증에 대한 강화를 하고, 접근 가능한 사용자를 명확히 해야 한다.

마지막으로 컨테이너 자원 고갈 위협을 방지하기 위해 운영체제는 실시간 모니터링 기능을 제공하여 자원 사용량을 추적하여 비정상적인 활동에 대해 알림을 설정하도록 해야 한다. 또한, 컨테이너의 자원 제한 플래그를 설정하여 각 컨테이너의 자원 사용량의 제한을 두어 적절한 자원 사용이 이루어지도록 해야 한다.

클라우드 환경에서의 운영체제는 컨테이너 탈출, 공급망 공격, 자원 고갈 등 다양한 사이버 위협에 대한 최전선의 방어선을 제공한다. 이를 위해 격리 메커니즘 강화, 이미지 검증 및 서명, 자원 사용량 제한, 실시간 모니터링과 같은 다양한 보안 전략이 필수적이다. 특히, 운영체제는 클라우드 환경에서의 동적이고 복잡한 위협을 효과적으로 감지하고 방지하기 위해 지속적으로 개선되고 있어야 한다. 이러한 다층적 보안 대책은 클라우드 환경의 안전성과 신뢰성을 유지하며, 안정적인 서비스 제공을 보장하는 데 핵심적인 역할을 한다.

3. 결론

3-1. 요약

1 장과 2 장을 통해 AI/ML, 빅데이터, 클라우드 시대에서의 운영체제의 역할과 존재하는 사이버 위협들, 그리고 운영체제 차원에서의 보호 및 지원을 확인할 수 있었다. 먼저 1 장에서는 운영체제의 개요를 살펴보고, 현대 사회에서의 운영체제의 역할을 알 수 있었으며, 운영체제 차원의 보안의 필요성을 명확히 했다. 이후 2 장에서는 구체적으로 각 분야에서 운영체제가 어떤 역할을 하는 지와 잠재하고 있는 수많은 사이버 위협들을 확인하고, 이를 대응할 수 있는 전략들을 구체적으로 알아볼 수 있었다.

AI/ML 환경에서는 고성능 연산과 대규모 데이터 처리를 지원하기 위해 GPU, 메모리, 입출력 자원 관리의 중요성이 강조되었으며, 데이터 유출, 모델 도난 및 변조, 데이터 중독 공격과 같은 위협에 대응하기 위한 다양한 보호 전략이 제시되었다. 빅데이터 환경에서는 대규모 데이터를 효과적으로 저장하고 처리하기 위한 분산 파일 시스템 관리와 작업 스케줄링이 운영체제의 주요 역할로 나타났다. 동시에 데이터 유출, 분산 노드의 취약점, DDoS 공격과 같은 위협들에 대한 대응 전략이 필요함을 확인했다. 클라우드 환경에서는 가상화 기술과 컨테이너 기술을 활용한 자원 격리와 동적 관리가 핵심적인 역할을 하며, 컨테이너 탈출, 공급망 공격, 자원 고갈 등 점점 증가하는 위협들을 방지하기 위한 보호 메커니즘이 요구되었다.

결론적으로, 현대 기술 환경에서 운영체제는 단순한 자원 관리 기능을 넘어, 다양한 기술 환경에서 보안과 안정성을 보장하며 기술적 도전과 위협에 대응하는 핵심적인 역할을 수행하고 있다. 이를 통해 운영체제는 AI/ML, 빅데이터, 클라우드 기술의 성공과 신뢰성을 뒷받침하는 필수적인 기반으로 자리 잡고 있음을 확인할 수 있었다. 앞으로의 운영체제는 이러한 역할을 더욱 강화하고, 지속적으로 변화하는 기술과 위협에 적응하며 발전해 나가야 할 것이다.

3-2. 운영체제 보안의 한계

앞서 살펴본 것에 따르면 운영체제를 통해 많은 것을 해결할 수 있었다. 물론 운영체제 차원에서의 보호를 통해 사이버 위협에 대한 초동 조치를 할 수 있는 것은 맞지만, 이에 대해 경각심을 가질 필요가 있다. 운영체제에 모든 보안적 책임을 맡긴다면 더 큰 위험을 초래할 수 있다. 고도로 발전되어 가고 있는 사이버 위협에서 운영체제의 보안 정책을 우회할 수 있는 공격들은 실제로 지속적으로 발전해 나가고 있으며 백도어와 같은 기술은 사용자가 인지하지 못한 채 지속적으로 공격을 수행해 나갈 수도 있는 문제다.

특히, 운영체제는 보안 메커니즘을 제공하지만 하드웨어, 소프트웨어, 네트워크 간의 복잡한 상호작용에서 모든 잠재적 위협을 완벽히 제거할 수는 없다. 하드웨어 취약점, 운영체제의 보안 취약점, 그리고 사용자 실수나 오용에 따른 위협은 운영체제의 통제를 벗어나는 경우가 많다. 예를 들어, 제로데이 공격은 운영체제의 보안 체계를 무력화할 수 있는 대표적인 사례로, 시스템이 완전히 보호되지 않는 상황에서 발생한다.

또한, 클라우드와 같은 분산 환경에서는 운영체제의 역할이 더욱 복잡해진다. 분산 환경의 특성상, 하나의 노드에서 발생하는 보안 취약점이 전체 시스템에 전파될 수 있다. 따라서 운영체제만으로는 분산 환경의 모든 보안 위협을 효과적으로 통제하기 어렵다. 이러한 한계는 운영체제가 다른 보안 기술 및 정책과 함께 통합적으로 작동해야 함을 보여준다.

마지막으로, 운영체제는 사용자의 인지적 한계와 의존성에도 영향을 받는다. 사용자는 보안 설정을 정확히 이해하거나 유지하지 못하는 경우가 많으며, 이는 악성 코드와 같은 위협을 발생시키는 주요 요인이 된다. 따라서 운영체제는 보안에 대한 전적인 책임을 지는 것이 아니라, 사용자 교육, 하드웨어 보안 기술, 네트워크 보안 등과의 협력을 통해 보완적으로 작동해야 한다.

결론적으로, 운영체제는 디지털 환경의 보안을 위한 강력한 첫 번째 방어선이지만, 그 자체로 완벽한 보안 솔루션은 될 수 없다. 기술의 발전과 사이버 위협의 정교화 속에서 운영체제는 지속적으로 진화하고 다른 보안 기술과 통합적으로 협력하며, 보다 강력하고 유연한 보안 체계를 구축해 나가야 할 것이다.

3-3. 미래 기술 발전에 따른 운영체제의 역할

현대 존재하는 AI/ML, 빅데이터, 클라우드 기술은 지속적으로 발전해 나갈 것이다. 이에 따라, 기술은 더욱더 복잡해지고, 변화되며, 신기술들이 등장할 것이 분명하다. 특히, 양자 물리학의 원리를 이용해 계산을 수행하는 양자 컴퓨팅의 등장과, 데이터를 중앙 클라우드에서 처리하는 것이 아닌 데이터 생성 지점에서 처리하는 기술인 엣지 컴퓨팅 기술이 도래하고 있다. 이러한 신기술들은 기존의 컴퓨팅 패러다임을 변화시키며, 운영체제 설계와 구현 방식에도 큰 도전 과제를 제시할 것이다.

양자 컴퓨팅의 경우, 기존의 이진 논리가 아닌 큐비트를 기반으로 작동하며, 기존 운영체제가 다루지 않았던 중첩과 얽힘 같은 물리적 특성을 지원해야 한다. 이는 양자 프로세서와 기존의 디지털 컴퓨터 간의 상호작용을 효율적으로 관리할 수 있는 새로운 운영체제 모델의 필요성을 의미한다. 또한, 양자 알고리즘의 실행을 최적화하고, 기존의 보안 체계를 뛰어넘는 양자 암호화 기술을 지원하는 운영체제 설계가 필수적일 것이다.

엣지 컴퓨팅 기술에서는 데이터가 생성되는 현장에서 실시간으로 처리되므로, 운영체제는 분산 환경에서의 데이터 처리, 네트워크 지연 최소화, 그리고 에너지 효율성 극대화를 중점으로 설계되어야 한다. 특히, 엣지 디바이스는 제한된 자원을 가지는 경우가 많아 경량화 된 운영체제와 자원 제한 환경에서 최적화된 스케줄링 및 메모리 관리가 필수적이다. 엣지 컴퓨팅과 클라우드 컴퓨팅의 하이브리드 환경을 지원하기 위한 새로운 프로토콜 및 보안 체계도 운영체제의 중요한 역할 중 하나로 대두될 것이다.

미래 기술의 발전은 운영체제에게 새로운 도전과 기회를 동시에 제공한다. 운영체제는 변화하는 기술 환경에 적응하며, 더 빠르고, 안전하며, 효율적인 시스템을 지원할 수 있는 방향으로 지속적으로 진화해야 한다. 이를 위해 하드웨어와 소프트웨어, 네트워크의 상호작용을 통합적으로 관리하며, 기술 발전 속도에 맞추어 새로운 아키텍처와 보안 메커니즘을 개발하고 도입할 필요가 있다.

결론적으로, 운영체제는 미래 기술의 핵심 기반으로서, 새로운 기술적 요구사항을 수용하고, 안전하고 안정적인 컴퓨팅 환경을 제공하는 데 중심적인 역할을 하게 될 것이다.

3-4. 마치며

에세이를 끝으로 운영체제, 고급모바일실험 2 를 마무리함과 동시에 한 학기에 대한 모든 과정이 끝이 났다. 한 학기 동안 느꼈던 점, 배운 점, 그리고 앞으로 나아갈 방향에 대해 생각하며 마무리해보고자 한다.

이번 학기는 대학 생활 중에 가장 힘들게 보냈던 학기라고 생각된다. 많은 양의 과제와 수업 내용에 대한 복습, 학부연구생 활동, 그리고 시험 기간에는 시험 공부까지 하루하루가 쉽지만은 않았던 것 같다. 늘 일정들을 확인하면서 '과연 기간 내에 끝낼 수 있을까?' 라는 생각을 많이 했던 것 같다. 그래도 항상 기간이 다가오면 어떠한 결과물이 생겨나 있는 것을 보면 신기하였다.

그만큼 바쁘게 지냈던 한 학기지만 고생한 만큼 보상은 달다는 말이 거짓말은 아닌 것 같다. 아직 많이 부족하지만 기존에 비해 레포트 작성 능력은 많이 성장했고, 학문을 바라보는 시야, 프로그래밍 능력은 정말 많이 성장했다고 느끼고 있다. 운영체제 수업은 그 중에서도 가장 큰 도전과 성장을 경험한 과목이었다. 이론적인 내용 또한 어려운데 코드로 구현하는 작업은 매번 쉽지 않았던 것 같다. 많은 자료를 찾아보고, 디버깅 작업도 많이 하며 늘 결과물을 만드려고 했던 것 같다. 결과물을 바탕으로 레포트를 작성하는 작업은 지겹기도 했지만 즐거운 작업이었다. 나의 생각을 글로 되풀이하는 작업은 귀찮을 수 있지만 남에게 나의 결과물을 보여주고, 그에 따른 성능을 보여주고 평가하는 작업이라 생각하니 즐겁게 할 수 있었다.

앞으로 더욱더 힘든 일들이 많을 것이고, 더욱 열심히 살아야 하는 것은 분명하다 생각된다. 그 시절이 오더라도 이번 학기를 생각하면 항상 해내었기에 긍정적인 생각과 이겨낼 수 있다는 삶의 태도를 가질 수 있을 것이다. 결론적으로, 어떤 역경이 다가와도 이겨낼 수 있다는 자신감을 얻었으며 학문에 대한 통찰력을 높이고, 실력적으로도 많이 성장할 수 있었던 한 학기였다고 생각된다.

끝으로 교수님께서는 학기 동안 매 강의마다 열정적으로 가르쳐 주시고, 강의 밖에서도 조언과 좋은 말씀을 해주셨던 기억이 있다. 교수님 덕분에 많은 것을 배우고 성장할 수 있었다고 생각된다. 에세이를 통해 한 학기 동안 최선을 다해주신 교수님께 감사 말씀을 드리며 에세이를 마친다.

한 학기 동안 정말 감사했습니다.