

1. **Microkernel**: In [computer science](#), a **microkernel** (often abbreviated as **μ-kernel**) is the near-minimum amount of [software](#) that can provide the mechanisms needed to implement an [operating system](#) (OS). These mechanisms include low-level [address space](#) management, [thread](#) management, and [inter-process communication](#) (IPC).

**Contents** [\[hide\]](#)

1	<a href="#">History</a>
2	<a href="#">Introduction</a>
3	<a href="#">Inter-process communication</a>
4	<a href="#">Servers</a>
5	<a href="#">Device drivers</a>
6	<a href="#">Essential components and minimality</a>
7	<a href="#">Performance</a>
8	<a href="#">Security</a>
9	<a href="#">Third generation</a>
10	<a href="#">Nanokernel</a>
11	<a href="#">See also</a>
12	<a href="#">References</a>
13	<a href="#">Further reading</a>

- [Kernel \(computer science\)](#)
  - [Hybrid kernel](#)
  - [Loadable kernel module](#)
  - [Monolithic kernel](#)
- [Microservices](#)
- [Tanenbaum–Torvalds debate](#)
- [Trusted computing base](#)
- [Unikernel](#)
- [Multi-Environment Real-Time](#)

2. **Software Portability**: **Portability** in [high-level computer programming](#) is the usability of the same [software](#) in different environments. The prerequisite for portability is the generalized [abstraction](#) between the application logic and [system interfaces](#). When software with the same functionality is produced for several [computing platforms](#), portability is the key issue for development cost reduction.

**Contents** [\[hide\]](#)

1	<a href="#">Strategies for portability</a>
1.1	<a href="#">Similar systems</a>
1.2	<a href="#">Different processors</a>
2	<a href="#">Source code portability</a>
2.1	<a href="#">Effort to port source code</a>
3	<a href="#">See also</a>
4	<a href="#">References</a>
5	<a href="#">Sources</a>

- [Cross-platform software](#)
- [Hardware-dependent Software](#)
- [C \(programming language\)](#)
- [Language interoperability](#)
- [Portability testing](#)
- [Source-to-source compiler](#)

3. MINIX 3: **Minix 3** is a project to create a small, [high availability](#), high functioning [Unix-like operating system](#). It is published under a [BSD license](#) and is a successor project to the earlier versions, [Minix 1](#) and [2](#).

Contents [hide]	
1	<a href="#">Goals of the project</a>
2	<a href="#">History</a>
3	<a href="#">Reliability policies</a>
3.1	<a href="#">Reduce kernel size</a>
3.2	<a href="#">Cage the bugs</a>
3.3	<a href="#">Limit drivers' memory access</a>
3.4	<a href="#">Survive bad pointers</a>
3.5	<a href="#">Tame infinite loops</a>
3.6	<a href="#">Limit damage from buffer overflows</a>
3.7	<a href="#">Restrict access to kernel functions</a>
3.8	<a href="#">Restrict access to I/O ports</a>
3.9	<a href="#">Restrict communication with OS components</a>
3.10	<a href="#">Reincarnate dead or sick drivers</a>
3.11	<a href="#">Integrate interrupts and messages</a>
4	<a href="#">Architecture</a>
5	<a href="#">Differences between MINIX 3 and prior versions</a>
6	<a href="#">Mascot</a>
7	<a href="#">MINIXCon</a>
8	<a href="#">See also</a>
9	<a href="#">References</a>
10	<a href="#">Further reading</a>
11	<a href="#">External links</a>

- [Comparison of operating system kernels](#)
- [MINIX file system](#)
- [List of computing mascots](#)
- [Category:Computing mascots](#)

4. My resident memory size at first is **728 KB**:

```
16710 erx      20  0  4340  728  656 T  0.0  0.0  0:00.00 a.out
```

After increasing the array by 10x and each malloc to 10,000, my new resident memory size is **5012KB**:

```
16889 erx      20  0 14072 5012 1016 T  0.0  0.2  0:00.00 a.out
```

The memory size is not 10x the old one.

5. After running the program, my resident memory size is **732 KB**:

```
7194 erx      20  0  4340  732  660 T  0.0  0.0  0:00.00 a.out
```

The resident memory was pretty similar to the previous one.

6. For the third test, I ran the program and my resident memory size is **640KB**:

```
17206 erx      20  0  4340  640  568 T  0.0  0.0  0:00.00 a.out
```

7. For the pre-test (without assigning), I got a resident memory size of **628KB**:

```
17515 erx      20  0  4340  628  556 T  0.0  0.0  0:00.00 a.out
```

After assigning the pointers, I got a resident memory size of **636KB**:

```
17485 erx      20  0  4340  636  568 T  0.0  0.0  0:00.00 a.out
```

After increasing the size, I got a resident memory size of **1824KB**:

```
17499 erx      20  0  5264 1824  996 T  0.0  0.1  0:00.00 a.out
```

After increasing the size, the size does increase (by around 3x).

8. Denial of service (DOS): In [computing](#), a **denial-of-service attack (DoS attack)** is a [cyber-attack](#) in which the perpetrator seeks to make a machine or network resource unavailable to its intended [users](#) by temporarily or indefinitely disrupting [services](#) of a [host](#) connected to the [Internet](#). Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.<sup>[1]</sup>

Contents <small>[hide]</small>	
1 History	4.17 SACK Panic
1.1 Hong Kong's Telegram	4.18 Shrew attack
1.2 <a href="#">Wikipedia down</a>	4.19 Slow Read attack
2 Types	4.20 Sophisticated low-bandwidth Distributed Denial-of-Service Attack
2.1 Distributed DoS	4.21 (S)SYN flood
2.2 Application layer attacks	4.22 Teardrop attacks
2.2.1 Application layer	4.23 Telephony denial-of-service (TDoS)
2.2.2 Method of attack	4.24 TTL expiry attack
2.3 Advanced persistent DoS	4.25 UPnP attack
2.4 Denial-of-service as a service	5 Defense techniques
3 Symptoms	5.1 Application front end hardware
4 Attack techniques	5.2 <a href="#">Application level Key Completion Indicators</a>
4.1 Attack tools	5.3 Blackholing and sinkholing
4.2 Application-layer floods	5.4 IPS based prevention
4.3 Degradation-of-service attacks	5.5 DDS based defense
4.4 Denial-of-service Level II	5.6 Firewalls
4.5 Distributed DoS attack	5.7 Routers
4.6 DDoS extortion	5.8 Switches
4.7 HTTP POST DoS attack	5.9 Upstream filtering
4.8 Challenge Collapsar (CC) attack	6 Unintentional denial-of-service
4.9 Internet Control Message Protocol (ICMP) flood	7 Side effects of attacks
4.10 Nuke	7.1 Backscatter
4.11 Peer-to-peer attacks	8 Legality
4.12 Permanent denial-of-service attacks	9 See also
4.13 Reflected / spoofed attack	10 References
4.14 Amplification	11 Further reading
4.15 Mirai botnet	12 External links
4.16 R-U-Dead-Yet? (RUDY)	

- Application layer DDoS attack
- BASHLITE
- Billion laughs
- Botnet
- Blaster (computer worm)

- Dendroid (malware)
- Fork bomb
- High Orbit Ion Cannon (HOIC)
- Hit-and-run DDoS
- Industrial espionage
- Infinite loop
- Intrusion detection system
- Low Orbit Ion Cannon (LOIC)
- Network intrusion detection system
- October 2016 Dyn cyberattack
- Paper terrorism
- Project Shield
- ReDoS
- Resource exhaustion attack
- SlowDroid
- Slowloris (computer security)
- UDP Unicorn
- Virtual sit-in
- Warzapping
- Web shell
- Wireless signal jammer
- XML denial-of-service attack
- Xor DDoS
- Zemra
- Zombie (computer science)

9. Keystroke Logging: **Keystroke logging**, often referred to as **keylogging** or **keyboard capturing**, is the action of recording (logging) the keys struck on a **keyboard**, typically covertly, so that person using the keyboard is unaware that their actions are being monitored. Data can then be retrieved by the person operating the logging program. A **keylogger** can be either **software** or **hardware**.

Contents <small>[hide]</small>	
1	<a href="#">Application</a>
1.1	<a href="#">Software-based keyloggers</a>
1.1.1	<a href="#">Keystroke logging in writing process research</a>
1.1.2	<a href="#">Related features</a>
1.2	<a href="#">Hardware-based keyloggers</a>
2	<a href="#">History</a>
3	<a href="#">Cracking</a>
3.1	<a href="#">Trojans</a>
3.2	<a href="#">Use by police</a>
4	<a href="#">Countermeasures</a>
4.1	<a href="#">Anti-keyloggers</a>
4.2	<a href="#">Live CD/USB</a>
4.3	<a href="#">Anti-spyware / Anti-virus programs</a>
4.4	<a href="#">Network monitors</a>
4.5	<a href="#">Automatic form filler programs</a>
4.6	<a href="#">One-time passwords (OTP)</a>
4.7	<a href="#">Security tokens</a>
4.8	<a href="#">On-screen keyboards</a>
4.9	<a href="#">Keystroke interference software</a>
4.10	<a href="#">Speech recognition</a>
4.11	<a href="#">Handwriting recognition and mouse gestures</a>
4.12	<a href="#">Macro expanders/recorders</a>
4.13	<a href="#">Deceptive typing</a>
5	<a href="#">See also</a>
6	<a href="#">References</a>
7	<a href="#">External links</a>

- [Anti-keylogger](#)
- [Black-bag cryptanalysis](#)
- [Computer surveillance](#)
- [Digital footprint](#)
- [Hardware keylogger](#)
- [Reverse connection](#)
- [Session replay](#)
- [Spyware](#)
- [Trojan horse](#)
- [Virtual keyboard](#)

10. Cryptography: **Cryptography** or **cryptology** (from [Ancient Greek](#): [κρυπτός](#), romanized: *kryptós* "hidden, secret"; and [γράφειν](#) *graphein*, "to write", or [-λογία](#) *-logia*, "study", respectively<sup>[1]</sup>) is the practice and study of techniques for [secure communication](#) in the presence of third parties called [adversaries](#).<sup>[2]</sup> More generally, cryptography is about constructing and analyzing [protocols](#) that prevent third parties or the public from reading private messages;<sup>[3]</sup> various aspects in [information security](#) such as data [confidentiality](#), [data integrity](#), [authentication](#), and [non-repudiation](#)<sup>[4]</sup> are central to modern cryptography. Modern cryptography exists at the intersection of the disciplines of [mathematics](#), [computer science](#), [electrical engineering](#), [communication science](#), and [physics](#). Applications of cryptography include [electronic commerce](#), [chip-based payment cards](#), [digital currencies](#), [computer passwords](#), and [military communications](#).

## Contents [hide]

- 1 Terminology
- 2 History of cryptography and cryptanalysis
  - 2.1 Classic cryptography
  - 2.2 Computer era
  - 2.3 Advent of modern cryptography
- 3 Modern cryptography
  - 3.1 Symmetric-key cryptography
  - 3.2 Public-key cryptography
  - 3.3 Cryptanalysis
  - 3.4 Cryptographic primitives
  - 3.5 Cryptosystems
- 4 Legal issues
  - 4.1 Prohibitions
  - 4.2 Export controls
  - 4.3 NSA involvement
  - 4.4 Digital rights management
  - 4.5 Forced disclosure of encryption keys
- 5 See also
- 6 References
- 7 Further reading
- 8 External links

- Outline of cryptography
  - List of cryptographers
  - List of important publications in cryptography
  - List of multiple discoveries
  - List of unsolved problems in computer science
- A Syllabical and Steganographical table – first cryptography chart
- Comparison of cryptography libraries
- Crypto Wars
- Encyclopedia of Cryptography and Security
- Global surveillance
- Information theory
- Strong cryptography
- W3C's Web cryptography API

11. Authentication: **Authentication** (from [Greek](#): αὐθεντικός *authentikos*, "real, genuine", from αὐθέντης *authentes*, "author") is the act of [proving](#) an [assertion](#), such as the [identity](#) of a computer system user. In contrast with [identification](#), the act of indicating a person or thing's identity, authentication is the process of verifying that identity. It might involve validating personal [identity documents](#), verifying the authenticity of a [website](#) with a [digital certificate](#),<sup>[1]</sup> determining the age of an artifact by [carbon dating](#), or ensuring that a product or document is not [counterfeit](#).

## Contents [\[hide\]](#)

- 1 [Methods](#)
- 2 [Authentication factors](#)
  - 2.1 [Single-factor authentication](#)
  - 2.2 [Multi-factor authentication](#)
- 3 [Authentication types](#)
  - 3.1 [Strong authentication](#)
  - 3.2 [Continuous authentication](#)
  - 3.3 [Digital authentication](#)
  - 3.4 [Product authentication](#)
    - 3.4.1 [Packaging](#)
- 4 [Information content](#)
  - 4.1 [Literacy and literature authentication](#)
- 5 [History and state-of-the-art](#)
- 6 [Authorization](#)
- 7 [Access control](#)
- 8 [See also](#)
- 9 [References](#)
- 10 [External links](#)

- [Access Control Service](#)
- [AssureID](#)
- [Atomic authorization](#)
- [Authentication Open Service Interface Definition](#)
- [Authenticity in art](#)
- [Authorization](#)
- [Basic access authentication](#)
- [Biometrics](#)
- [CAPTCHA](#)
- [Chip Authentication Program](#)
- [Closed-loop authentication](#)
- [Diameter \(protocol\)](#)
- [Digital identity](#)
- [EAP](#)
- [Electronic authentication](#)
- [Encrypted key exchange \(EKE\)](#)
- [Fingerprint Verification Competition](#)
- [Geolocation](#)
- [Hash-based message authentication code](#)
- [Identification \(information\)](#)
- [Java Authentication and Authorization Service](#)
- [Kantara Initiative](#)
- [Kerberos](#)
- [Multi-factor authentication](#)
- [Needham–Schroeder protocol](#)
- [OAuth – an open standard for authorization](#)
- [OpenAthens](#)
- [OpenID Connect – an authentication method for the web](#)

- [OpenID](#) – an authentication method for the web
- [Provenance](#)
- [Public-key cryptography](#)
- [RADIUS](#)
- [Reliance authentication](#)
- [Secret sharing](#)
- [Secure Remote Password protocol \(SRP\)](#)
- [Secure Shell](#)
- [Security printing](#)
- [SQRL](#)
- [Strong authentication](#)
- [Tamper-evident technology](#)
- [TCP Wrapper](#)
- [Time-based authentication](#)
- [Two-factor authentication](#)
- [Usability of web authentication systems](#)
- [Woo–Lam](#)

12. Biometrics: **Biometrics** is the technical term for body measurements and calculations. It refers to metrics related to human characteristics . Biometrics authentication (or realistic authentication)<sup>[note 1]</sup> is used in computer science as a form of identification and [access control](#).<sup>[1][2]</sup> It is also used to identify individuals in groups that are under [surveillance](#).<sup>[3]</sup>

Contents <a href="#">[hide]</a>	
1	<a href="#">Biometric functionality</a>
2	<a href="#">Multimodal biometric system</a>
3	<a href="#">Performance</a>
4	<a href="#">History</a>
5	<a href="#">Adaptive biometric systems</a>
6	<a href="#">Recent advances in emerging biometrics</a>
6.1	<a href="#">Operator signatures</a>
6.2	<a href="#">Proposed requirement for certain public networks</a>
6.3	<a href="#">Animal biometrics</a>
6.4	<a href="#">Video</a>
7	<a href="#">Issues and concerns</a>
7.1	<a href="#">Surveillance humanitarianism in times of crisis</a>
7.2	<a href="#">Human dignity</a>
7.3	<a href="#">Privacy and discrimination</a>
7.4	<a href="#">Danger to owners of secured items</a>
7.5	<a href="#">Presentation attacks</a>
7.6	<a href="#">Cancelable biometrics</a>
7.7	<a href="#">Soft biometrics</a>
7.8	<a href="#">International sharing of biometric data</a>
7.9	<a href="#">Likelihood of full governmental disclosure</a>
8	<a href="#">Countries applying biometrics</a>
8.1	<a href="#">India's national ID program</a>
9	<a href="#">See also</a>
10	<a href="#">Notes</a>
11	<a href="#">References</a>
12	<a href="#">Further reading</a>
13	<a href="#">External links</a>

- [Aadhaar](#)
- [Access control](#)
- [AFIS](#)
- [AssureSign](#)
- [BioAPI](#)



- Biometric passport
- Biometric voter registration
- Biometrics in schools
- BioSlimDisk
- Facial recognition system
- Fingerprint recognition
- Fuzzy extractor
- Gait analysis
- Government database
- Hand geometry
- Handwritten biometric recognition
- Identity Cards Act 2006
- International Identity Federation
- Iris recognition
- Keystroke dynamics
- Multiple Biometric Grand Challenge
- Private biometrics
- Retinal scan
- Signature recognition
- Smart city
- Speaker recognition<sup>[84]</sup>
- Surveillance
- Vein matching
- Voice analysis