

# Spis treści

<b>1</b>	<b>System Operacyjny</b>	<b>1</b>
1.1	Co robi? . . . . .	1
1.2	Jaką rolę pełni system operacyjny? . . . . .	1
1.3	Zasoby . . . . .	1
1.4	Wywołania systemowe . . . . .	1
1.5	Tryb jądra . . . . .	2
1.6	Rozruch komputera . . . . .	2
<b>2</b>	<b>Procesor</b>	<b>2</b>
2.1	Cykl rozkazowy . . . . .	2
2.2	Przerwania . . . . .	2
2.2.1	Przerwania zegarowe . . . . .	2
2.3	Rejestry . . . . .	3
2.4	Wielowątkowość . . . . .	3
2.5	Potoki . . . . .	3

## 1 System Operacyjny

System operacyjny jest warstwą oprogramowania operującą bezpośrednio na sprzęcie, której celem jest zarządzanie zasobami systemu komputerowego i stworzenie użytkownikowi środowiska łatwiejszego do zrozumienia i wykorzystania.

### 1.1 Co robi?

- Pośredniczy między programami a sprzętem
- Tworzy środowisko w którym operują programy użytkowników (low level interfejsy)
- Steruje komputerem i nadzoruje jego działanie
- Zarządza dostępem do zasobów

### 1.2 Jaką rolę pełni system operacyjny?

- System operacyjny dostarcza abstrakcji programom
- System operacyjny zarządza wszystkimi składnikami złożonego systemu
- System operacyjny zapewnia kontrolowany i uporządkowany przydział zasobów

### 1.3 Zasoby

- Czas procesora
- Pamięć operacyjna
- System plików
- Urządzenia wejścia wyjścia
- ...

### 1.4 Wywołania systemowe

Wywołania systemowe oddają kontrolę systemowi operacyjnemu za prośbą procesu i obsługują np.: dostęp do plików, tworzenie wątku itp.

## 1.5 Tryb jądra

Wszystkie współczesne systemy operacyjne rozróżniają tryb jądra i użytkownika. W trybie jądra operacje oraz dostęp przez procesor do zasobów nie jest ograniczony. Jest to w pewnym sensie jak najniższa warstwa jaką może osiągnąć kod. Aby operować na tej warstwie kod musi być częścią systemu operacyjnego. W trybie użytkownika dostęp kodu do zasobów jest kontrolowany przez system operacyjny.

## 1.6 Rozruch komputera

1. BIOS weryfikuje urządzenia i ilość pamięci
2. BIOS określa urządzenie rozruchowe z pamięci CMOS
3. BIOS wczytuje program rozruchowy
4. Program rozruchowy wczytuje system operacyjny
5. System operacyjny odczytuje listę urządzeń z konfiguracji BIOS i sterowniki

## 2 Procesor

Współczesne procesory operują zgodnie z architekturą von Neumanna. Ta architektura zakłada:

- Program jak i dane znajdują się w pamięci operacyjnej
- Rozkazy wykonuje się dokładnie w tej kolejności w jakiej znajdują się w pamięci (odstępstwa: instrukcje skoku, wywołania, powroty)
- Procesor przechowuje w rejestrze adres komórki pamięci zawierającej kolejną do wykonania instrukcję. W celu pobrania tej instrukcji, procesor wystawia odpowiedni adres na magistrali adresowej.

### 2.1 Cykl rozkazowy

1. Pobranie kodu rozkazu
2. Pobranie operandów
3. Składowanie wyniku
4. Rozpoznania przerwania

### 2.2 Przerwania

- Przerwania pozwalają na zatrzymanie cyklu rozkazowego procesora, dzięki czemu system operacyjny może odzyskać kontrolę lub zareagować na zdarzenie asynchroniczne
- Przerwania mogą pochodzić z trzech źródeł:
  - Zewnętrzne – I/O, zegary, układy bezpośredniego dostępu do pamięci
  - Diagnostyczne – w wyniku wykrycia nieoczekiwanego zdarzenia
  - Programowe – w wyniku wykonania instrukcji przerwania (niezależnie od trybu)
- Procesor może decydować czy przyjąć nowe przerwanie podczas obsługi innego, lub wyłączyć obsługę przerw zupełnie
- Kolejność obsługi przerw kontroluje kontroler przerw

#### 2.2.1 Przerwania zegarowe

Zwykle czasomierz generuje około 100 przerw na sekundę. Umożliwiają one systemowi operacyjnemu analizę stanu systemu i reakcję na zmiany

## 2.3 Rejestry

Rejestr to znajdująca się niedaleko procesora pamięć. Procesor wykorzystuje je do przechowywania danych oraz wyników operacji. W momencie zmiany kontekstu system operacyjny zachowuje stan rejestrów w pamięci operacyjnej.

## 2.4 Wielowątkowość

Dzięki wielowątkowości procesor może przechować stan procesora, i potem do niego wrócić. W ten sposób możemy przechowywać stan kilku procesów (wątków), dając iluzję jednoczesności z wystarczająco szybkim przełączaniem. Wielowątkowość też pozwala zachować czas procesora w momentach w których jakiś wątek czeka.

## 2.5 Potoki

Współczesne procesory dzięki specjalnych architekturach mogą jednocześnie wykonywać etapy cyklu rozkazowego. Np. w trakcie wykonywania polecenie pobrać kod nowego polecenia