

Techniki komputerowe w robotyce

WYKŁAD OSTATNI

Błędy systemów komputerowych – źródła i klasyfikacja

Robert Muszyński
KCiR, W4, PWr

– Skład Foil \TeX –

© R. Muszyński 2012-2015

Błędy systemów komputerowych

Błąd – właściwość programu lub sprzętu komputerowego powodująca nieprawidłowe działanie

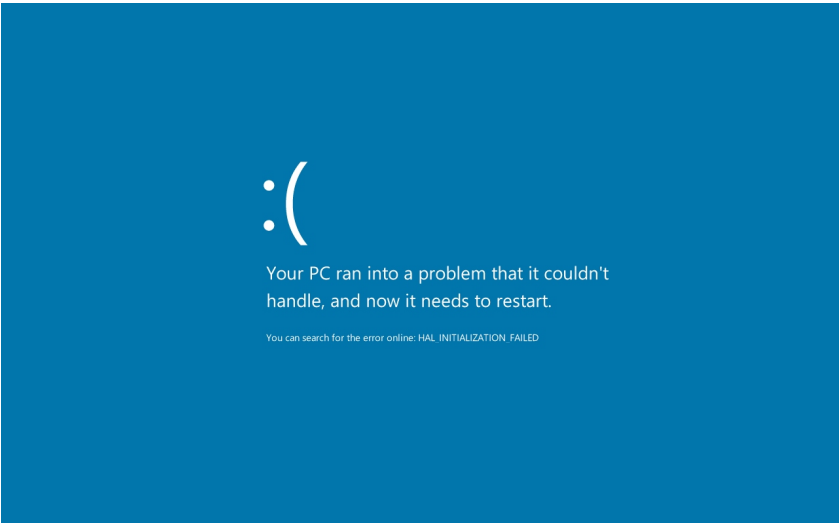
```
*** STOP: 0x00000404 (0x00000000,0x12345678,0x69696969,0x97f86a65)
PAGE_NOT_FOUND*** ADDRESS 09034672 has base at 09037000 - abort

CPUID:GenuineIntel Pentium Irql:1f SYSVER: 0xf0096404

Dll Base DateStmp Name Dll Base DateStmp Name
80100000 336546ff - ntoskrnl.exe 80010000 331054d1 - iexplore.exe
80037386 33104f49 - atapi.sys 80070000 3341a4d9 - outlook.exe
88013faa 33107e98 - epsy.sys 8021bf09 3342ab4f - disk.sys
80037389 3310354d - CLASS2.SYS 80102848 334013fb - Ntfs.sys
80037390 3310098d - Siowid.sys f95cb000 3340101f - Ntfs.sys
800ae122 3310h896 - Floppy.sys f1035800 336735af - Parport.sys
800bf392 33100f89 - KSecDD.sys f031fde0 34dc3ccf - Beep.sys
801cca04 3310h854 - i8042prt.sys f95cb000 3340101f - mouclass.sys
80037398 331034fg - kbdc.lass.sys f3e3b000 33104f49 - ctrl2cap.sys
80037399 3310f987 - UIDEOPORT.SYS f42cb000 3340101f - msfs.sys
80037400 3310g653 - vga.sys f63cb640 3340101f - iexplore.exe
80037401 33106g89 - npfs.sys

Address dword dump Build [13811 - Name
c0948732 3340101f 8021bf09 80037391 12345678 80037401 80a0b00f - Ntfs.sys
80948732 80070000 80037386 12345678 8021bf09 1c005638 df00eabf - rnl.exe

Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option. If this message reappears,
Contact your system administrator or technical support group.
```



Dążenie do opracowania systemu bezbłędnego – mrzonka

Błędy systemów komputerowych

Wymagamy by system komputerowy był:

- poprawny
- niezawodny
- bezpieczny

Poprawność – stała własność systemu, polegająca na tym, że system jest zgodny z jego specyfikacją

Niezawodność – stopień w jakim można się spodziewać, że system realizuje planowane funkcje z wymaganą precyzją

Bezpieczeństwo – zespół zasad, jakimi należy się kierować projektując i wykorzystując system komputerowy, tak aby poprawnie i w całości realizował on tylko i wyłącznie cele zgodne z intencjami właściciela

Poprawność a niezawodność

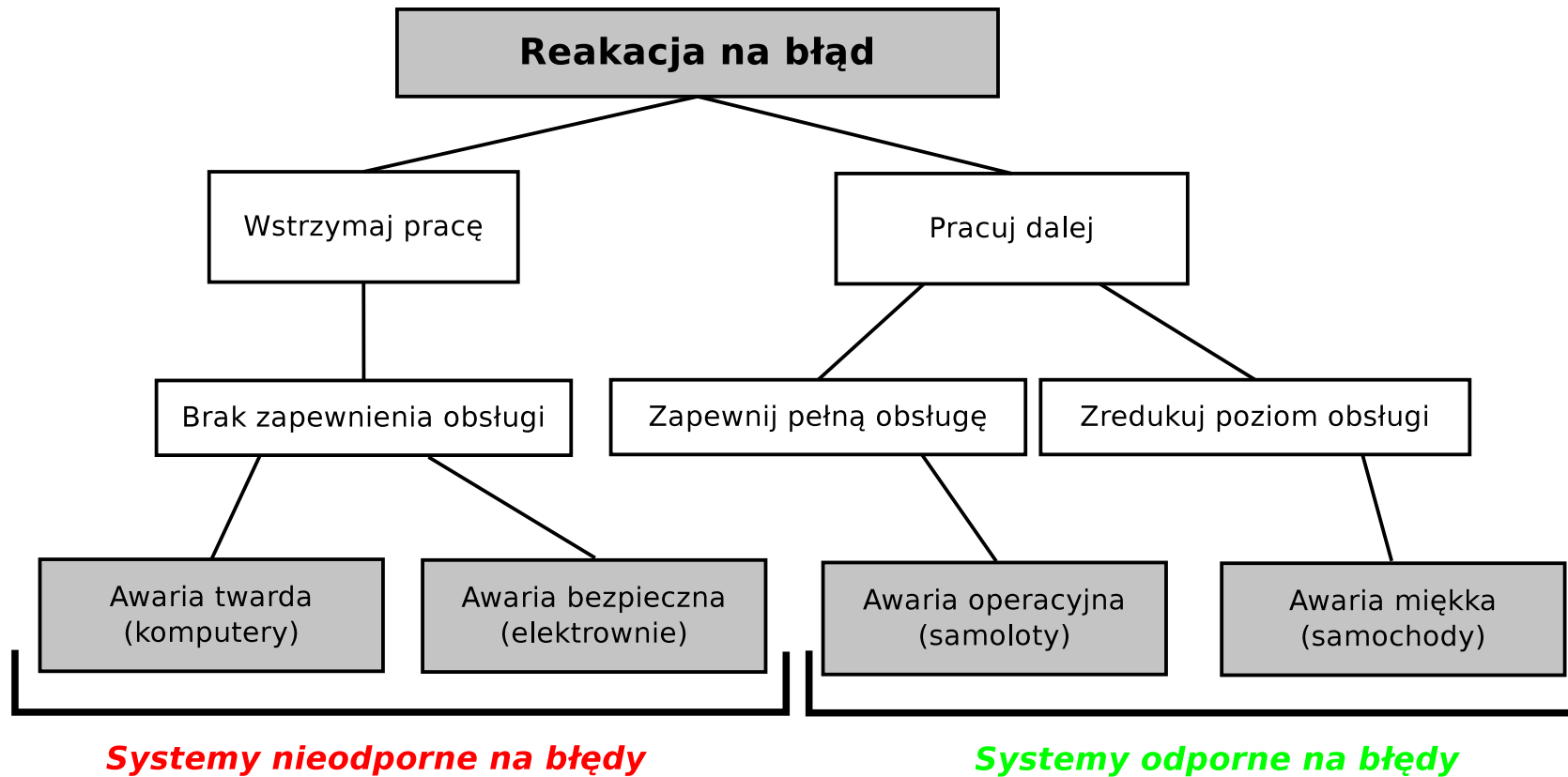
Czy system uznany za poprawny może nie być niezawodny i czy system niepoprawny może być niezawodny?

1. System sprawdzony w warunkach laboratoryjnych nie działa w miejscu docelowym
2. Źle napisany system mimo to działa niezawodnie, bo nadrzędne jego warstwy korygują działania źle napisanych warstw podrzędnych

Bezpieczeństwo a niezawodność

System w 100% bezpieczny, może być w 0% niezawodny i vice versa

Sposoby obsługi błędów



Klasyfikacja błędów

Błędy systemu:

- na etapie projektowania
- na etapie programowania
 - syntaktyczne
 - semantyczne
 - logiczne
 - algorytmiczne
- wynikające z wpływu otoczenia

Przykłady błędów

- Na etapie projektowania
 - maszty radarowe na angielskich niszczycielach
 - mechanizm zwalniania rakiet w F-18
 - HMS Sheffield – Falklandy-Malwiny
 - USS Vincennes z system AEGIS i irański Airbus A300
 - F-16 vs rakiety Patriot
- Na etapie programowania – błędy składni
 - użycie niepoprawnego symbolu (`var c:integer:`)
 - niepoprawne użycie symbolu (`x=y` zamiast `x==y`)
- Na etapie programowania – błędy znaczenia
 - niewłaściwie zrozumiana specyfikacja
 - niewłaściwie realizowana specyfikacja

Zasada

Czym mniej napiszesz, tym mniej popełnisz błędów –
dobór właściwego języka programowania

Przykłady błędów

- Na etapie programowania – błędy logiczne
 - pętle z warunkami post-check zamiast pre-check
 - przypadkowe pętle nieskończone
 - niezainicjowane zmienne
 - zakleszczenia (deadlock)
 - źle sformułowane warunki logiczne
- Na etapie programowania – błędy algorytmiczne
 - przekroczenie zakresu liczb
 - przekroczenie precyzji
 - nieuwzględnienie zakresu wartości obsługiwanych przez urządzenia we/wy
- Wynikające z wpływu otoczenia

Sprzęt \longleftrightarrow Oprogramowanie \longleftrightarrow Człowiek

Ranking błędów (według MITRE CWE/SANS)

25. Wyścig (race condition)
24. Używanie złych algorytmów kryptograficznych
23. Otwarte przekierowania (np. phishing)
22. Nielimitowana alokacja zasobów
21. Złe uprawnienia do krytycznych zasobów
20. Pobieranie niesprawdzonego kodu
19. Brak uwierzytelnienia przy krytycznych funkcjach systemu
18. Błąd w obliczeniach długości bufora
17. Przepelnienie wartości liczby całkowitej
16. Ujawnianie informacji w komunikatach błędów
15. Zła obsługa nadzwyczajnych lub warunkowych zdarzeń/wartości
14. Zła walidacja wskaźnika
13. Brak kontroli nazwy pliku dla polecenia Include/Require w języku PHP

12. Dostęp do bufora z niewłaściwą długością
11. Osadzenie uwierzytelnienia w oprogramowaniu
10. Brak szyfrowania wrażliwych danych
9. Brak kontroli parametrów przekazywanych do systemu operacyjnego
8. Brak ograniczeń typów plików, które można umieścić na serwerze
7. Brak ograniczeń ścieżki do konkretnego katalogu (path traversal)
6. Poleganie na niezaufanych informacjach przy uwierzytelnieniu
5. Błędy w przyznawaniu uprawnień
4. Cross-site Request Forgery (CSRF)
3. Kopiowanie danych bez kontroli długości (przepełnienie bufora) – klasyk!
2. Niewłaściwa kontrola parametrów używanych do zapytania SQL
1. Brak zachowania struktury serwisu webowego (Cross-site Scripting)