helion.pl
KSIĘGARNIA INTERNETOWA · WYDAWNICTWO

IDŹ DO
PRZYKŁADOWY ROZDZIAŁ
SPIS TREŚCI

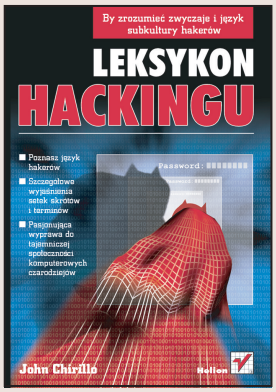
KATALOG KSIĄŻEK
KATALOG ONLINE
ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK
DODAJ DO KOSZYKA

CENNIK I INFORMACJE
ZAMÓW INFORMACJE O NOWOŚCIACH
ZAMÓW CENNIK

CZYTELNIA
FRAGMENTY KSIĄŻEK ONLINE

Leksykon hackingu



Autor: John Chirillo
Tłumaczenie: Andrzej Grażyński
ISBN: 83-7361-283-1
Tytuł oryginału: Hack Attacks Encyclopedia
Format: B5, stron: 192

Hakerzy pojawili się, gdy tylko pierwsze komputery trafiły na uczelnie. W przeciwieństwie do rasowych informatyków, którzy właściwie nie potrzebowali komputerów do swoich teoretycznych rozważań, hakerzy starali się wykorzystać sprzęt praktycznie. Wkrótce stali się prawdziwymi czarodziejami, znającymi od podszewki zasady działania maszyn, na których pracowali. Niewielka część z nich zajęła się włamaniami do systemów, większość poświęciła się pożytecznym zajęciom, dzięki którym powstało wiele cennych i używanych do dziś programów, a nawet języków programowania.

Hakerzy tworzyli i tworzą rozproszoną społeczność, która wykształciła własne zwyczaje, a także swój własny, tajemniczy i trudny do zgłębienia dla profanów język. Jemu właśnie poświęcona jest ta książka. Dzięki niej poznasz slang, którym posługują się hakerzy. Wiele z ukutych przez nich terminów wyszła poza wąskie środowisko hakerskie i używana jest dzisiaj przez szerokie grono osób, często w nieco zmienionym znaczeniu. Dzięki tej książce poznasz, co znaczyły one oryginalnie i przeżyjesz pasjonującą podróż po świecie hakerów.

Setki skrótów i określeń, w których zawarta jest historia komputerów i internetu, czeka na Ciebie. Nie wiesz, co to jest „płeć bajtowa”? Sięgnij po tę książkę, a wszystko stanie się łatwe.

- Poznasz język hakerów,
- Szczegółowe wyjaśnienia setek skrótów i terminów,
- Pasjonująca wyprawa po tajemniczej społeczności komputerowych czarodziejów.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Wstęp **5**

Leksykon hackingu **7**

Leksykon hackingu

20x
patrz *TOPS-20*

abbrev
Skrót od wyrazu *abbreviation* — „skrót".

ABEND
(*ABnormal END*) Nienormalne zakończenie programu; *crash*, *lossage*. Oryginalnie nazwa makra systemów operacyjnych OS i DOS komputerów IBM/360 i IBM/370 z lat 1960 – 1980, powodującego natychmiastowe zakończenie wykonywanego programu z inicjatywy systemu operacyjnego lub użytkownika. Z reguły pisane w całości wielkimi literami (ABEND — jak we wspomnianych systemach), spotykane także w postaci *abend*. Nieświadomi etymologii tego słowa hakerzy kojarzą go z niemieckim określeniem wieczoru (niem. *abend*) — ma ono nazywać czynności operatorów systemu, którzy późnym piątkowym popołudniem wprowadzają system operacyjny w stan tzw. pracy bezobsługowej na czas weekendu.

ACK
1. Skrót oznaczający potwierdzenie (ang. *acknowledge*); mnemoniczne określenie znaku ASCII(6) używanego (w telegrafii) w charakterze kodu potwierdzającego transmisję. Używany także w odniesieniu do reakcji programu interakcyjnego (np. *ping*) na zapytanie użytkownika. **2**. Uprzejme potwierdzenie tego, iż zrozumiało się punkt widzenia dyskutanta („Ack. Ack. Ack. Teraz już wiem" — rzadko stosowane w komunikacji polskojęzycznej). **3**. Prośba o potwierdzenie obecności partnera („Jesteś tam?") przy komputerze podczas rozmowy w trybie terminalowym (jak w programie *talk*). **4**. Symboliczne ponowienie zapytania zawartego w poprzednio wysłanym e-mailu, który pozostał bez odpowiedzi; zobacz także *NAK*.

Ada
Wywodzący się z Pascala język programowania, który stanowi podstawę dla projektów informatycznych w Departamencie Obrony USA. Według zgodnej opinii hakerów język ten jest koronnym przykładem tego, czego dokonać może rozbudowana biurokracja: stworzony został przez specjalny komitet, jest niespójny koncepcyjnie, trudny w użytkowaniu, a jego stworzenie kosztowało miliardy

dolarów (z tego względu określany bywa żartobliwie jako „język *PL/I* lat osiemdziesiątych"). Szczególnie wyśmiewany jest sposób obsługi wyjątków oraz komunikacja między procesami. Nazwa języka kojarzona jest z imieniem córki Lorda Byrona, Ady Lovelace; jej korespondencja z Charlesem Babbage'em w połowie XIX wieku może być (ze względu na treść i charakter) uważana za pierwszy, historycznie udokumentowany przykład programowania maszyny, oczywiście w rozumieniu ówczesnych, dziewiętnastowiecznych realiów — „program" był wówczas niczym innym, jak tylko odpowiednim sprzężeniem niezliczonych kół zębatych, zapadek itp. Najbardziej elegancka analogia, jaką sformułowano pod adresem języka Ada, to „słoniątko"; patrz także *słoniowe programy* — mały, poczciwy język, który aż prosi się, by „wypuścić go na wolność".

adger

Czasownik uknuty w kręgach Uniwersytetu Kalifornijskiego w Los Angeles określający czynności o daleko idących negatywnych konsekwencjach (jak na przykład nieodwracalne wykasowanie ważnych plików), którym można było zapobiec przy odrobinie rozsądku.

ad-hackery, ad-hockery, ad-hocity

1. Nieuzasadnione pochlebstwo w stosunku do programu (np. systemu eksperckiego), który przejawia co prawda pozory inteligentnego zachowania, lecz wynika ono z zastosowania bardzo prostych mechanizmów; na przykład program kwestionujący te z wprowadzanych słów, które nie znajdują się w jego słowniku może być (przez nieświadomego użytkownika) podejrzewany o „zdolność" rozumienia języka naturalnego. **2**. Specjalny rodzaj kodu, przeznaczony do obsługi („wygładzania") pewnych nietypowych danych wejściowych, które „normalnie" doprowadziłyby do załamania się programu; patrz także *efekt ELIZY*.

admin

Skrót często używany (zwłaszcza w korespondencji *online*) na oznaczenie osoby sprawującej opiekę nad systemem („administratora"). Skrótami pokrewnymi są: *sysadmin* (określający osobę odpowiedzialną za konfigurację systemu operacyjnego), *siteadmin* (na określenie administratora witryny www) i *newsadmin* (na określenie moderatora grup dyskusyjnych); zobacz także *postmaster*, *sysop*, *system mangler*.

ADVENT

Prototyp komputerowej gry przygodowej, zaimplementowany pierwotnie przez Willa Crowthera na komputerze PDP-10 (jako próba „zaprogramowania" fantazji i wyobraźni) i rozwinięty przez Dona Woodsa do postaci gry-układanki (*puzzle*). Obecnie gra ta znana jest bardziej pod nazwą *Adventure*; zobacz także *vadding*. Definiuje ona zwięzły, by nie powiedzieć oschły, nieco naiwny styl charakterystyczny dla gier tekstowych („Dziki, zielony wąż właśnie przepełznął Ci drogę", „Nie widzę tu…", „Jesteś właśnie w labiryncie, którego wszystkie zakręty są identyczne"). Pewne „magiczne słowa" w rodzaju *xyzzy*, *plugh* itd. mają prawdopodobnie swe źródło w opisywanej grze.

Przy okazji, Crowther uczestniczył w eksploracji systemu jaskiń Mamooth & Flint Ridge; „posiada" on obecnie jaskinie Colosal Cave i Bedquilt („kołdra") oraz Y2 — to ostatnie oznacza w żargonie speleologów „tylne wejście".

AIDS

Akronim utworzony od ang. *A Infected Disk Syndrome* — syndrom zainfekowanego dysku, przy czym litera „A" kojarzona jest często z komputerami Apple. Analogia do znanego medycznego zespołu nabytych niedoborów immunologicznych (*Acuired Immune Deficiency Syndrome*) ma akcentować zagrożenie ze strony nierozważnej wymiany danych między komputerami, porównywalnej z niebezpiecznym seksem; patrz także *wirus*, *robak*, *trojan*, *system dziewiczy*

AI-zupełne problemy

Termin stworzony w MIT, używany na określenie (przez analogię do problemów *NP-zupełnych*) problemów (oraz podproblemów) ściśle związanych ze sztuczną inteligencją (AI), dla rozwiązania których niezbędny jest dość spory zasób inteligencji właściwej istotom rozumnym; innymi słowy, problemy AI-zupełne są zbyt trudne, by można je było całkowicie rozwiązać wyłącznie za pomocą komputera.

Przykładami problemów AI-zupełnych są: mechaniczne postrzeganie i interpretacja obrazów w sposób właściwy człowiekowi oraz rozumienie języka naturalnego.

akronimy rekursywne

Wywodząca się z MIT hakerska koncepcja tworzenia akronimów odwołujących się do samych siebie bądź grupy akronimów odwołujących się do siebie nawzajem. Najbardziej znane akronimy tego rodzaju związane są z edytorami używanymi w MIT (EINE — *Eine Is Not Emacs*, ZWEI — *Zwei Was Eine Initially*). Istnieje także kompilator języka Scheme, zwany LIAR (*Liar Imitates Apply Recursively*); patrz też *GNU* (w znaczeniu **1**.) — *GNU's Not Unix*. Nazwę firmy CYGNUS objaśnia się natomiast jako *Cygnus, Your GNU Support*; patrz także *EMACS*.

aktywne oczekiwanie

patrz *zajęty czekaniem*

akumulator

1. Archaiczne określenie jednego z rejestrów uniwersalnych procesora, traktowanego przez niektóre rozkazy w sposób specjalny. W procesorach firmy Intel rejestrami takimi są: *AL*, *AX* i *EAX*. W dawnych komputerach akumulator był w zasadzie jedynym rejestrem (w dzisiejszym znaczeniu tego słowa); w stosunku do obecnych procesorów ich rejestry określane są za pomocą oznaczeń symbolicznych, a słowo „akumulator" nie jest w ogóle używane. **2**. Rzadko już używane określenie służące odróżnieniu rejestru używanego (przez program) na potrzeby operacji arytmetycznych i logicznych od rejestrów związanych z adresowaniem, indeksowaniem lub zliczaniem obrotów pętli („W tej pętli sumującej używane elementy akumulatorem jest *EDX*"). **3**. W odniesieniu do tzw. maszyn bezadresowych, w których podstawowe znaczenie odgrywały operacje *stosowe*, określenie „akumulator" używane było niekiedy jako synonim *stosu*.

alfa cząstki

patrz *rozpad bitowy*

aliasowania błąd

Subtelny błąd programowania związany z dynamicznym przydziałem pamięci (dokonywanym m.in. przez funkcje *New* i *GetMem* w Pascalu lub *malloc* w C).

W sytuacji, gdy na przydzielony obszar pamięci wskazuje kilka wskaźników („aliasów"), zwolnienie tego obszaru za pomocą jednego z nich powoduje, że pozostałe stają się tzw. *wiszącymi wskaźnikami*, czyli wskaźnikami o niepoprawnej zawartości. Istotą błędu aliasowania jest właśnie odwoływanie się do tychże wiszących wskaźników.

Receptą na błąd aliasowania jest przede wszystkim unikanie tworzenia wielu wskaźników do jednego obszaru (nie zawsze jest to możliwe); błąd ten nie występuje w systemach zwalniających nieużywaną pamięć w sposób automatyczny (patrz *odśmiecanie*), na przykład w językach *Lisp* i Java.

all-elbows
(ang. „rozpychanie się łokciami") Określenie programu rezydentnego (TSR — *Terminate and Stay Resident*), korzystającego z zasobów systemowych w sposób egoistyczny, bez respektowania obecności innych programów rezydentnych. Jeżeli w systemie istnieje program rezydentny przechwytujący przerwania klawiatury, obecność innego „egoistycznego" programu rezydentnego może spowodować zablokowanie klawiatury.

ALT
1. Klawisz na klawiaturze komputerów PC i kompatybilnych. **2**. Klawisz polecenia na komputerach Macintosh. **3**. Alternatywna nazwa klawisza ESC (kod ASCII 27) na terminalu komputera PDP-10.

alt-bit
patrz *metabit*

Aluminiowa Księga
Żartobliwe określenie podręcznika *Common Lisp: The Language*, autorstwa G. L. Steele'a (wyd. Digital Press 1984, 1990). W rezultacie korekty edytorskiej, niektóre fragmenty tekstu w drugim wydaniu wydrukowane zostały w kolorze przypominającą metaliczną zieleń (*yucky green* wg określenia autora); patrz także *Błękitna księga*, *Księga Kopciuszka*, *Księga Diabelska*, *Księga Smoka*, *Zielona księga*, *Pomarańczowa księga*, *Księga różowej koszulki*, *Purpurowa księga*, *Czerwona księga*, *Srebrna księga*, *Biała księga*, *Księga czarodzieja*, *biblia*.

ameba
Humorystyczne określenie komputera osobistego Commodore Amiga.

Amoeba
1. Rozproszony, heterogeniczny system operacyjny opracowany w latach 1981 – 1983 na Uniwersytecie Vrije w Amsterdamie pod kierunkiem A. Tanenbauma. **2**. Patrz *ameba*.

amp off
Synonim uruchomienia procesu w *tle*; nazwa pochodzi od uniksowego operatora & (*ampersand*).

amper
Skrótowe określenie *ampersanda.*

ampersand

&, znak ASCII o kodzie 38.

AOS

1. (przestarzałe) Synonim inkrementacji (zwiększenia) czegokolwiek, w analogii do rozkazu inkrementacji komputera PDP-10 (*Add One and do not Skip* — „Dodaj jedynkę i nie przeskakuj"); „zwiększenie" może być rozumiane w szerszym sensie, np. dołożenie drewna do ogniska. Dlaczego litera „S" symbolizować ma tu frazę *don not Skip*, choć intuicyjnie kojarzy się z czymś wręcz przeciwnym — *Skip*? Odpowiedzi na to pytanie należy poszukiwać w folklorze związanym z PDP-10. W architekturze tego komputera istnieje osiem rozkazów łączących inkrementację, jej wynik oraz ew. przeskok, w zależności od wyniku, między innymi:

- *AOSE* — (*Add One and then Skip next instruction if result is Equal to zero*) Dodaj 1 i, jeżeli wynikiem dodawania jest zero, przeskocz następny rozkaz;
- *AOSG* — (*Add One and then Skip next instruction if result is Greater than zero*) Dodaj 1 i, jeżeli wynik dodawania jest dodatni, przeskocz następny rozkaz;
- *AOSN* — (*Add One and then Skip next instruction if result is Not equal to zero*) Dodaj 1 i, jeżeli wynikiem dodawania nie jest zero, przeskocz następny rozkaz;
- *AOSA* — (*Add One and then Skip Always*) Dodaj 1 i bezwarunkowo przeskocz następny rozkaz;

itd. Czwarta litera skrótu mnemotechnicznego rozkazu określa zatem warunek, przy spełnieniu którego należy przeskoczyć następny rozkaz. Jeżeli więc dany rozkaz nie ma w ogóle wykonywać przeskoku, jego mnemotechniczny skrót pozbawiony jest czwartej litery.

Analogicznie wygląda sprawa z rozkazem *skoku* AOJ (*Add One and do not Jump* — „Dodaj jedynkę i nie wykonuj skoku").

Oto inne dziwactwo PDP-10: rozkaz *SKIPx*, gdzie *x* oznacza jedną z ośmiu liter, jest rozkazem przeskoku następnej instrukcji w przypadku spełnienia warunku określonego przez tę piątą literę — na przykład *SKIPA* oznacza przeskok bezwarunkowy. Rozkaz, który bezwarunkowo nie wykonuje przeskoku, powinien więc mieć mnemonikę *SKIP* — na przekór znaczeniu intuicyjnemu! Identycznie rozkaz skoku *JUMP* — *JUMPA* oznacza skok bezwarunkowy, zaś *JUMP* — brak skoku.

Jedną z osobliwości architektury PDP-10 było również to, że rozkaz wykonujący skok i odtwarzający wybrane znaczniki (*JRST* — *Jump and ReSTore flag*) wykonywał się szybciej od rozkazu *JUMPA*; określenie pustego zbioru znaczników powodowało więc efekt identyczny ze skokiem bezwarunkowym, tyle że szybciej wykonywanym — z czego hakerzy skwapliwie korzystali.

Wprawdzie komputery PDP-10 należą już do gatunku wymarłego, warto jednak pamiętać, do jakich zawiłości doprowadzić może niefrasobliwie stosowana mnemotechnika.

2. System operacyjny wywodzący się z systemu *Multics*, wspierany swego czasu przez firmę Data General. **3**. *Algebraic Operating System* — określenie algebraicznej

notacji *wrostkowej*, czyli takiej, w której operator znajduje się *między* operandami, dla odróżnienia od notacji *przyrostkowej* (w której operator występuje po argumentach) i *przedrostkowej* (w której operator poprzedza argumenty). Określenie *Operating System* stanowi żartobliwe nawiązanie do kalkulatorów, których większość stosuje notację algebraiczną; patrz także *notacja polska odwrotna*.

aplikacja

patrz *app*

app

Skrót od *aplikacji* (programu użytkowego), dla odróżnienia od programu systemowego. Hakerzy przejawiają tendencję do innego klasyfikowania programów, zaliczając do oprogramowania systemowego kompilatory, systemy komunikatów, edytory, a nawet niektóre gry — mimo że „normalny" użytkownik skłonny byłby uważać je za aplikacje.

arc wars

(z ang. „wojny archiwizerów") Niekończące się kłótnie o to, który z archiwizerów jest lepszy, a dokładniej — spór o wyższości archiwizera *arc* (firmy SEA) i *PKarc* (firmy PKware). *PKarc* był nie tylko szybszy i dostarczał lepszej kompresji, lecz także umożliwiał zachowanie wstecznej kompatybilności formatu archiwum z *arc*. Aby uniknąć wysokich kosztów licencyjnych, firma *PKware* zmuszona była do zmiany nazwy swego archiwizera na *PKpak*. W miarę, jak na rynku zaczęły pojawiać się coraz lepsze archiwizery, wykorzystujące zaawansowane algorytmy kompresji, lecz niekompatybilne z *arc*, ten ostatni stracił całkowicie swe znaczenie.

archiwizator

Program tworzący archiwum z grupy plików; intencjonalnie archiwum ma mieć mniejszy rozmiar od sumarycznego rozmiaru archiwizowanych plików, choć (co można udowodnić matematycznie) dla każdego archiwizatora istnieje taki plik (lub grupa plików), dla którego efekt archiwizacji będzie wręcz odwrotny. Pierwszym znanym archiwizatorem był DOS-owy *arc* firmy SEA (System Enhancement Associates). Obecnie do najpowszechniejszych należą ZIP i RAR.

archiwum

1. Efekt działania *archiwizatora*. **2**. W popularnym znaczeniu — grupa plików dostępna do ściągnięcia na serwerze FTP.

arena

Obszar pamięci przydzielony do procesu uniksowego za pomocą *brk*(*2*) oraz *sbrk*(*2*) i używany przez *malloc*(*3*) jako pamięć dynamiczna.

arg

Skrót na oznaczenie parametru („argumentu") funkcji, procedury lub makra.

ASCII

(*American Standard Code for Information Interchange* — amerykański standardowy kod wymiany informacji) Pierwowzór kodu używanego na potrzeby dzisiejszych komputerów. Oryginalnie kod ASCII zawierał 128 znaków, kodowanych na 7 bitach. W przeciwieństwie do wcześniejszych kodów, kod ASCII rozróżniał małe i wielkie

litery, nie było w nim jednak miejsca na znaki „narodowe" charakterystyczne wyłącznie dla konkretnych języków. Z biegiem czasu znaki te znalazły swe miejsce w „górnej połówce" rozszerzonego, 8-bitowego kodu ASCII, co jednak nie przysporzyło mu cech „kodu uniwersalnego", na przekór usiłowaniom niektórych producentów sprzętu i oprogramowania; patrz także *EBCDIC*.

ASCII art

patrz *Sztuka ASCII*

atak siłowy

(ang. *brute force*) Określenie prymitywnego stylu programowania, wykorzystującego raczej ogromną moc obliczeniową komputera niż inteligencję swego twórcy. Podejście takie zazwyczaj ignoruje skalę problemu, w wyniku czego metody przydatne do analizowania niewielkich problemów używane są do rozwiązywania (czy raczej próby rozwiązywania) problemów o nieporównywalnie większej skali.

Jeden z przykładów ataku siłowego związany jest z tzw. problemem komiwojażera (ang. TSP — *Travelling Salesman Problem*) stanowiącym przykład problemu NP-trudnego. Komiwojażer wyrusza z pewnego miasta w celu odwiedzenia *N* wybranych miast i następnie powrotu do punktu wyjścia; sumaryczna długość przebytej trasy ma być jak najmniejsza. „Siłowe" podejście do problemu polega na generowaniu *wszystkich możliwych* tras i sprawdzaniu ich długości. Jedyną zaletą takiego algorytmu jest łatwość jego zaprogramowania; oprócz generowania najbardziej nawet absurdalnych tras (wszak sprawdzane są wszystkie) algorytm ten ma podstawową wadę, jaką jest jego złożoność, proporcjonalna do *N!* („N silnia") — dla N równego 15 liczba tras równa jest 1 307 674 368 000; dla N = 1000 jest ona niewyobrażalna.

Innym, prostszym koncepcyjnie przykładem „programowania siłowego" jest sortowanie listy (za pomocą istniejącej procedury sortującej) w celu znalezienia jej najmniejszej liczby.

To, czy „programowanie siłowe" zasługuje na ignorancję, czy też jest praktyką godną polecenia, zależy od konkretnego przypadku. W przypadku niewielkich problemów nawet kilka godzin dodatkowego czasu komputera nie jest czasem straconym, jeżeli oznacza oszczędność kilku dni (czy tygodni) czasu programisty, który ten musiałby poświęcić na stworzenie algorytmu rozwiązującego problem w ciągu minuty. Ten „bardziej" inteligentny algorytm prawdopodobnie byłby algorytmem bardziej złożonym, a więc stwarzającym większe ryzyko popełnienia błędu w porównaniu z mało efektywnym, lecz prostszym algorytmem „siłowym".

Ken Thompson, współtwórca systemu UNIX, jest autorem słynnego powiedzenia: „W razie wątpliwości zastosuj *brute force*". Mimo na pozór żartobliwego charakteru tego stwierdzenia, nie sposób nie dojrzeć tkwiącego w nim głębszego sensu — algorytm „siłowy", jako prostszy, jest raczej bezbłędny, a więc daje wiarygodne wyniki. Nie można także zapominać, iż to właśnie prostota, solidność i przenośność systemu UNIX przyczyniły się głównie do jego popularności. Wśród wielu kompromisów, na jakie nieustannie skazywani są twórcy oprogramowania, wybór między algorytmem „siłowym" a algorytmem bardziej „inteligentnym" uwarunkowany jest wieloma czynnikami natury technicznej, ekonomicznej i estetycznej.

attoparsek

W układzie jednostek SI przedrostek *atto* oznacza jedną trylionową (10^{-18}), parsek jest natomiast jednostką długości równą ok. 3,26 roku świetlnego. Będący iloczynem tych wielkości *attoparsek* równy jest ok. 3.1 cm, tak więc attoparsek/*microfortnight* to prędkość ok. 1 cala na sekundę. Jednostka ta używana jest oficjalnie (choć może nie do końca poważnie) przez hakerów w Wielkiej Brytanii.

avatar

Alternatywna do *root* nazwa *superużytkownika* na niektórych komputerach z systemem UNIX. Jej autorstwo przypisuje się jednemu z hakerów, który przeszedł z firmy CMU do Tektronixa.

awk

1. Interpretowany język do obróbki danych tekstowych, wynaleziony przez Alfreda Aho, Petera Weinbergera i Briana Kernighana (nazwa języka pochodzi od pierwszych liter ich nazwisk). Charakteryzuje się składnią zbliżoną do języka C, swobodą używania zmiennych (bez konieczności ich deklarowania) i tablic skojarzeniowych oraz zorientowanym na pola przetwarzaniem tekstu; patrz także *Perl*. **2**. Skrót od ang. *awkward* („niezgrabny") używany zwłaszcza w odniesieniu do wyrażeń z trudem poddających się przetwarzaniu na podstawie wyrażeń regularnych (np. wyrażeń zawierających znaki nowego wiersza).

azbest

Środki lub działania chroniące przed *płomieniami*. Obecnie najczęściej mają one formę działań piętnujących, polegających na przyznawaniu kompromitujących nagród (*asbestos cork award* — nagroda azbestowego korka) lub obdarowywaniu szczególnymi detalami odzieżowymi (*asbestos longjohn*).

backdoor

(ang. dosł. „tylne drzwi") Luka w mechanizmach bezpieczeństwa systemu, pozostawiona przez twórców lub administratorów. Luka taka nie zawsze jest wynikiem zaniedbania i nie zawsze stanowi zagrożenie dla bezpieczeństwa; niekiedy jest ona konieczna dla zapewnienia właściwego nadzoru ze strony personelu serwisowego lub wsparcia technicznego dostawcy. Luki takie mają jednak to do siebie, że administratorzy zwyczajnie o nich zapominają, dzięki czemu pozostają one w systemach dłużej, niż ktokolwiek by się spodziewał. Niesławnej pamięci robak RTM, który jesienią 1988 roku wyrządził niemałe szkody w komputerach z systemem UNIX BSD, wykorzystywał lukę zabezpieczenia w programie *sendmail*.

Artykuł Kena Thompsona na łamach czasopisma *Communications of the ACM 27*, *8* z sierpnia 1984 roku opisuje poważną lukę we wczesnych wersjach UNIX-a, którą można uznać za jedno z najbardziej spektakularnych zaniedbań pod względem bezpieczeństwa. Kompilator języka C zawierał mianowicie kod rozpoznający, w którym miejscu znajduje się procedura logowania i wprowadzał kod wrażliwy na hasło wybrane przez Thompsona; dawało to Thompsonowi możliwości „wejścia" do systemu nawet wówczas, gdy nie miał on w tym systemie swego konta.

Luka tego rodzaju powinna być naprawiona poprzez usunięcie niebezpiecznego fragmentu z kodu źródłowego kompilatora i jego ponowne skompilowanie. To jednak prowadziło do powstania błędnego koła, bowiem „spreparowana" wersja

kompilatora zawsze wstawiała kwestionowany kod do produkowanych binariów, które nadal pozostawały „spreparowane". W efekcie rzeczona luka nadal pozostawała w kodzie binarnym, choć *nie było już jej śladu w kodzie źródłowym*!

backgammon

patrz *Bignum, moby, pseudopierwsza liczba*

backspace and overstrike

(ang. dosł. „cofnij i przekreśl") Sygnał, iż pewna wypowiedź jest błędna i należy ją zweryfikować.

backward compatability

(zniekształcenie od *backward compatibility* — „kompatybilność wstecz") Niezgodność nowej wersji sprzętu lub oprogramowania z protokołami, formatami czy geometrią złącz, z którymi zgodne były poprzednie wersje; zarzucenie obsługi starszych mechanizmów (koncepcji) na rzecz nowszych i bardziej efektywnych; patrz także *dzień flagowy*.

BAD

(ang. *broken as designed* — „marny zgodnie z projektem") Określenie programu, którego nieprzydatność, bezużyteczność, niefunkcjonalność nie jest wynikiem błędów, lecz złego zaprojektowania.

bag on the side

(ang. w wolnym tłumaczeniu „torba na ramieniu") Modyfikacja oryginalnego produktu, która w założeniu miała mu przydać funkcjonalności, lecz wydaje się bardziej kłopotliwa niż użyteczna („C++? To tylko bagaż ramionach *C*").

bagbiter

1. Cokolwiek — na przykład program lub komputer — co charakteryzuje się częstymi awariami lub funkcjonuje w sposób dziwaczny (na przykład edytor tekstu ograniczający długość linii do 80 znaków). **2**. Osoba, która — umyślnie lub nie — przyczyniła się do czyichś kłopotów, na przykład wskutek dostarczenia błędnego programu.

Najstarszym — i jak na razie jedynym znanym — programem, który celowo miał opisane cechy był program Lexiphage w laboratorium AI instytutu MIT. Program ten na wybranym terminalu produkował (stylizowanymi literami) napis „THE BAG" w otoczeniu pożerających go szczęk.

bajt

Jednostka pamięci lub danych, o rozmiarze umożliwiającym reprezentowanie jednego znaku alfabetycznego. We współczesnych komputerach bajt ma zwykle 8-bitów (to wyraz dążenia do nadawania rzeczom rozmiarów będących potęgą liczby 2), lecz np. w maszynie z 36-bitowym słowem bajty są z reguły 9-bitowe[1]. Komputer PDP-10 zapewnia obsługę bajtów, które są w istocie *polami bitowymi* o długości od 1 do 36.

[1] W popularnych niegdyś maszynach serii ODRA-1300 w 24-bitowym słowie można było pomieścić 4 znaki 6-bitowe, a repertuar rozkazów maszyny zawierał kilka instrukcji umożliwiających dostęp do poszczególnych znaków — *przyp. tłum.*

Określenie powstało w 1956 roku, we wczesnej fazie projektu komputera IBM Stretch i oznaczało jednostkę 6-bitową (ówczesne urządzenia wejścia-wyjścia posługiwały się 6-bitowymi porcjami transmisji). Bajty ośmiobitowe ukształtowały się ostatecznie w projekcie komputera IBM/360 w końcu roku 1956; patrz także *półbajt*.

bamf

1. Charakterystyczny dźwięk wydawany przez teleportowany obiekt lub postać w grze komputerowej, szczególnie w ramach *rzeczywistości wirtualnej*. **2**. Dźwięk towarzyszący magicznym transformacjom w grach rzeczywistości wirtualnej. **3**. Akronim od *Bad-Ass Mother F***er*, używany na określenie jednego z populacji nieprzyjemnych potworków w grach LPMUD (i podobnych).

bananowa etykieta

Określenie etykiet przylepianych na bokach szpulki taśmy magnetycznej; kształt każdej z tych etykiet przypomina tępo zakończony banan.

bananowe zjawisko

patrz *HAKMEM*, nr 176

bananowy problem

Problem z właściwym określeniem warunku zatrzymania iteracji (przez analogię do małej dziewczynki, która stwierdziła „Umiem już wymawiać słowo *banana*, tylko nie wiem, kiedy skończyć"); patrz także *błąd ogrodzenia*, *Dissociated Press*, *HAKMEM* (nr 176).

banner

1. Strona tytułowa dodawana do wydruku przez program obsługi kolejki wydrukowej (*spooler*), zawierająca zwykle nazwę i identyfikator użytkownika, wypisane stylizowanymi, powiększonymi znakami (stworzonymi w grafice znakowej — patrz *sztuka ASCII*). Zwana także stroną rozdzielającą, łatwo bowiem wskazuje miejsce oddzielenia wydruków należących do różnych użytkowników. **2**. Podobna strona tytułowa, drukowana na kilku sąsiednich stronach „składanki" przez wyspecjalizowany program, np. UNIX-owy program *banner*(*1,6*). **3**. Pierwszy ekran wyświetlany przez program interaktywny, zawierający zwykle logo autora i informację na temat praw autorskich.

bar

Druga pod względem popularności nazwa (po *foo*, przed *baz*) na liście nazw zmiennych metasyntaktycznych („Załóżmy, że mamy dwie funkcje: FOO i BAR; FOO wywołuje BAR…"). **2**. Często łączone z *foo*, co daje w wyniku *foobar*.

BartleMUD

Każda z gier mających swój pierwowzór w oryginalnej grze MUD Richarda Bartle'a. Każdy z BartleMUD-ów charakteryzuje się niekonwencjonalnym humorem, surową, lecz przyjazną dla użytkownika, składnią poleceń i brakiem przymiotników w opisach obiektów. Niektórzy zwolennicy MUD-ów wolą unikać konotacji z nazwiskiem Bartle'a, lansując w zamian nazwę „MUD-1".