

# Sapos Piramidais nas guerras virtuais

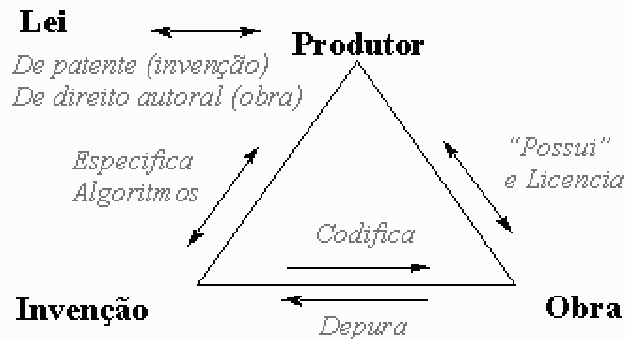
## Episódio IV: O Caso SCO versus IBM, Linux

Palestra no 4o. Forum Internacional do Software Livre  
Pontifícia Universidade Católica - Porto Alegre - RS, 6/06/03  
Prof. Pedro Antonio Dourado de Rezende  
Departamento de Ciência da Computação  
Universidade de Brasília

---

Aos desavisados, caiu como uma bomba uma notícia de 23 de Maio, de que a SCO havia entrado com uma ação judicial contra a IBM, no valor de "pelo menos um bilhão de dólares", por quebra de contrato de uso do Unix, envolvendo o Linux. A IBM teria reusado importantes partes do código do Unix no Linux, ambos sistemas operacionais por ela distribuídos. Este reuso estaria proibido no contrato, além de violar a propriedade intelectual que a empresa alega ter sobre o Unix system 5. A IBM nega, enquanto a SCO envia carta a mais de 1500 empresas que usam o GNU/Linux ameaçando-as com a possibilidade de futuras ações indenizatórias.

## O software como propriedade intelectual

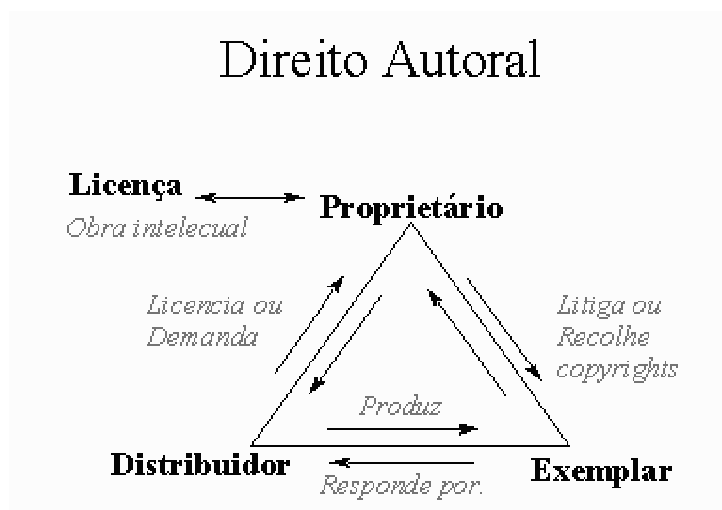


As surpresas vêm do fato da SCO, antes Caldera International, ter sido uma das primeiras empresas a ganhar dinheiro distribuindo GNU/Linux, e da contundência de suas ameaças, em face da ambiguidade em torno da propriedade intelectual que afirma ter. Em matéria de 29 de maio no Jornal do Commercio do Rio, o CEO da SCO é citado: *"Temos o direito contratual de impedir o uso impróprio de código, métodos e conceitos do Unix no Linux"*.

Que contrato é esse, e o que vem a ser "métodos e conceitos do Unix"? Trata-se de um contrato pelo qual a SCO afirma ter adquirido da Novell em 1995 a propriedade intelectual do Unix system 5. A Novell, por sua vez, os havia adquirido em 1992 da AT&T, sua criadora. Mas a opinião da Novell sobre o que foi comprado e vendido é diferente. Seu CEO é citado na mesma matéria afirmando: *"o acordo de 1995 não passou os direitos associados à SCO"*.

O bate-boca prossegue: *"É claro que somos donos dos direitos e podemos exigir o respeito a eles e às patentes"* disse o CEO da SCO, mesmo admitindo ser esse contrato *"um pouco confuso"*. Enquanto o CEO da Novell pressiona: *"Queremos que a SCO retire as suas afirmativas falsas a respeito da propriedade de patentes do Unix ou forneça informações conclusivas a respeito"*.

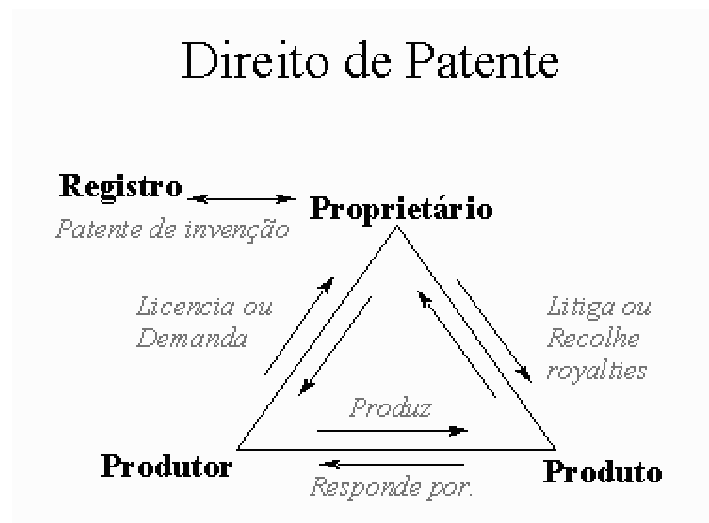
Já na Alemanha, onde o GNU/Linux ganha cada vez mais penetração no mercado e força nos corações e mentes dos internautas, a empresa LinuxTag entrou na justiça com um pedido de liminar, obrigando a SCO a apresentar evidências de que o Linux viola seus direitos ou retirar suas ameaças, sob pena de multas diárias. A liminar foi concedida em 4 de junho, e a única reação da SCO foi tirar do ar seu site em alemão, contendo cópia da carta-ameaça. A exposição de motivos da LinuxTag dizia: "*A SCO não pode prejudicar seus competidores com alegações levianas, com ameaças a seus clientes e com difamações à reputação do GNU/Linux impunemente*"



Para entendermos o que está por trás disso tudo, e quais as implicações para o futuro da informática e da sociedade informatizada, precisamos conhecer um pouco da história do Unix e daquilo que pode ser considerado propriedade intelectual do software. O nó da questão é a diferença entre direito autoral, que protege o código-fonte de um programador contra uso não autorizado, e patente, que protege um inventor com o direito de exploração industrial exclusiva da sua invenção. O fato desses direitos serem temporários não diz muito, haja vista as extensões aprovadas quando há interesse.

## Parte II

"Métodos e conceitos" de programação de computadores só se tornaram patenteáveis a partir do primeiro pedido não recusado, depois de incontáveis recusas, nos EUA em 1981. É que a lei americana veda a concessão de patentes sobre "leis da natureza", e como tal se costuma considerar, além de bem comum, as construções lógico-linguísticas. Além disso, leis de patentes também exigem que o invento seja inédito, útil e não óbvio. De lá para cá, entretanto, como numa corrida pelo ouro alquímico da revolução digital, foram concedidas mais de cem mil patentes desse tipo, por burocratas cuja noção de utilidade, obviedade e ineditismo no mundo dos bits tem se mostrado estranha, para dizer o mínimo.



Conhecidas como "patentes de software", são outorgas de monopólio através de descrições herméticas, genéricas e abstratas o suficiente para lançar advogados de bolsos fundos em verdadeira caça aos bruxos da programação -- principalmente os autônomos --, enquadrando como propriedade alheia praticamente qualquer idéia imaginável do que possa ser feito através de software. Antes que se considere exagerada esta afirmação, seria bom conhecer detalhes sobre o processo de concessão e litígio de tais patentes.

Tomemos o caso emblemático protagonizado por Leon Stambler contra a RSA, que Bruce Schneier comenta em sua

revista eletrônica Chrypto-gram de 15 de março (<http://www.counterpane.com/>). Munido de duas patentes sobre protocolos de autenticação digitais, Stambler vinha extorquindo empresas que comercializam software para segurança na informática. A análise técnico-jurídica das centenas de páginas de suas patentes custaria às vítimas mais caro do que as centenas de milhares de dólares que ele pedia, em troca da suspensão das ameaças.

Quando ameaçou a RSA por violação das suas patentes no SSL, a vítima resolveu trugar. A RSA havia submetido seis patentes do SSL em 1994 nos EUA, concedidas em 1997, enquanto as de Stambler haviam sido submetidas em 1992 e concedidas em 1999. Patentes são retroativas à data de submissão, apesar do processo transcorrer em sigilo. Quem submete o pedido é obrigado a divulgar apenas um resumo do pedido, que não pode ser modificado durante o trâmite da concessão.

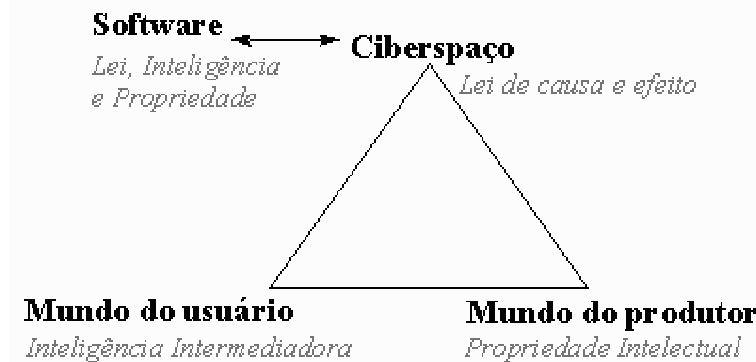
Mas o truque é que o pedido em si pode ser modificado enquanto tramita, requerendo nova análise -- sigilosa -- a cada modificação. É a chamada patente "submarino". Leo Stambler teve sete anos para cozinhar suas centenas de páginas de legalês em sigilo, enquanto descobria as idéias que a RSA e outros vinham conseguindo patentear, e que poderiam se encaixar no resumo genérico do seu pedido. E os tratados de "livre comércio" querem estender a jurisdição dessas patentes a todo o mundo.

Para mostrar a transparência da roupa nova do rei, um advogado australiano pediu recentemente o registro de patente do seu invento, que, depois de concedida, anunciou ele tratar-se da patente da roda. (<http://edition.cnn.com/2001/WORLD/asiapcf/auspac/07/02/australia.wheel/>). "Mas isso invalida a patente!", desculpa-se a autoridade australiana, enquanto acusa o advogado de má fé. Sim, mas e os tais "métodos e conceitos" de programação descritos em legalês? Por acaso os bits fazem no chip alguma coisa que não possa fazer o lápis no papel, não importa quão morosamente? O que há de novo sob o sol no que tange a

gerência de filas, de arquivos e de recursos, com o advento dos sistemas operacionais, além da linguagem e dos parâmetros?

Para os avisados, era só questão de tempo até que uma tal quizumba -- e seus desdobramentos dos tribunais -- atingisse o software livre, que não tem cacife nem vocação para essa banca de jogo. Há muito Richard Stallman, fundador da Free Software Foundation (FSF), vem disso nos avisando. O caso SCO versus Linux é o primeiro tiro da verdadeira revolução digital, onde degladiam por supremacia jurídica, de um lado, a liberdade do capital para auto-organizar-se, e do outro, a liberdade humana para buscar o conhecimento. Em suma, a disputa pela supremacia na virtualíssima trindade de Lessig.

## Software como ideologia: A virtualíssima trindade de Lessig



### Parte III

Quem detém, afinal, a posse dos "métodos e conceitos do Unix"? O UNIX originou-se do MULTICS. A homepage do MULTICS (<http://www.multicians.org/history.html>) o descreve como "um projeto multi-institucional inicialmente de pesquisa, que gerou grande influência no desenvolvimento dos sistemas operacionais". O projeto, iniciado pelo MIT, AT&T e GE, prometia, na infância da

computação digital em 1965, a revolucionária "programação interativa", com terminais substituindo pilhas de cartões perfurados.

O primeiro sistema operacional MULTICS, derivado do sistema CTSS do MIT, em operação em 1961, entrou em uso em 1969, ainda incompleto e instável, depois de quatro anos perseguindo uma especificação "móvel", deslizando em sua ambição. Neste mesmo ano a AT&T abandonou o projeto, cujos pesquisadores envolvidos passaram a desenvolver um sistema alternativo no laboratório de pesquisa da empresa, o Bell Labs. Este sistema alternativo veio a ganhar o nome de marca Unix, entrando em uso em 1973, depois de reescrito na linguagem C -- que com ele nascia -- por Ken Thompson e Dennis Ritchie.

Na ocasião a AT&T era concessionária de monopólio de telefonia nos EUA, e as regras desta concessão não lhe permitiam comercializar software. Devido ao interesse das instituições acadêmicas num sistema operacional inovador e útil ao ensino da programação, a AT&T fez acordos com várias universidades, a partir de 1975, pelos quais o código-fonte do Unix era cedido, por uma taxa nominal e sem nenhum compromisso de suporte, e autorizadas as universidades a fazerem atualizações, correções e modificações por conta própria, algumas das quais vieram a ser incorporadas em versões posteriores do Unix pela AT&T.

Em 1978 um dos autores do Unix -- Thomposon -- passou um ano como professor na Universidade de Berkeley, onde ele e seus alunos escreveram a primeira versão de um sistema semelhante, que ficou conhecido como BSD (Berkeley System Distribution). O BSD evoluiu para os sistemas free BSD, que por sua vez, tendo ganho dos regentes da UC Berkeley licenças open source, passaram a ser livremente distribuídos (ver "Twenty years of Berkeley unix", <http://www.oreilly.com/catalog/opensources/book/kirkmck.html>).

Foi só em 1983 que a AT&T recebeu permissão legal para vender licenças de software. A sua versão do Unix de então passou a ser comercializada como "System V", nome que a distingue das versões adaptadas por outras empresas sob outras licenças. Em 1992 a AT&T vendeu os direitos do System V para a Novell, que, por sua vez entrou em litígio com a Universidade de Berkeley no ano seguinte, alegando haver o BSD reusado código-fonte do Unix system V. A disputa se resumiu a três arquivos dentre cerca de 18 mil. O caso foi resolvido amigavelmente, com a reescrita de boa parte do BSD durante o ano de 1994, antes que pudesse voltar a ser distribuído, mas com a Novell insistindo que o acordo fosse mantido em segredo de justiça.

Berkeley reescreveu boa parte do código que estava em ambos sistemas, mas não todo. Por que? O detalhe nesse ponto é a insistência do departamento jurídico da Universidade para que o acordo permitisse a manutenção de algum código-fonte idêntico no freeBSD e no System V. Por que Berkeley insistiu, e por que foi nisso atendida, mantendo-se algum código duplicado no módulo de controle de processos de ambos sistemas operacionais, quando do arquivamento da disputa? Esta pergunta é importante, pois tem relação com a possibilidade de má fé na litigação da SCO.

Por ocasião do litígio entre a Novell e a Universidade de Berkeley, nenhuma patente dos "métodos e conceitos" do Unix foi cogitada. Mesmo porque, se as houvessem, poderiam ser questionadas em relação ao ineditismo, devido à sua origem em projeto de pesquisa científica -- o MULTICS -- cujas publicações, em seis artigos no Fall Joint Computer Conference de 1965 e nas mais de três mil páginas do MSPM, sacramentavam tais conceitos e métodos em domínio público. A disputa entre a Novell e a Universidade de Berkeley se resumiu, portanto, ao copyright. Isto é, aos direitos autorais sobre o código-fonte do Unix system V.

## **Parte IV**



Na negociação que pôs fim ao litígio entre Berkeley e Novel envolvendo copyright do Unix system V, Berkeley insistiu na manutenção de algum código duplicado para que ficasse caracterizado a via de mão dupla na prévia cooperação entre a AT&T e várias universidades, permitindo e promovendo a contribuição cruzada entre as várias versões de sistemas operacionais que surgiram da primeira distribuição de código-fonte do Unix. Esses sistemas vieram a ser conhecidos como "unix-like", tendo sua disseminação, evolução e suporte se dado através de grupos virtuais como o AUUG e o USENIX, das quais o System V havia também se beneficiado (<http://minnie.cs.adfa.oz.au/Seminars/AUUG96/pdppaper.html>).

Veja, por exemplo, um dos vários diagramas de influência entre esses sistemas, aqui mostrado apenas aqueles de linhagem relevante para o caso:

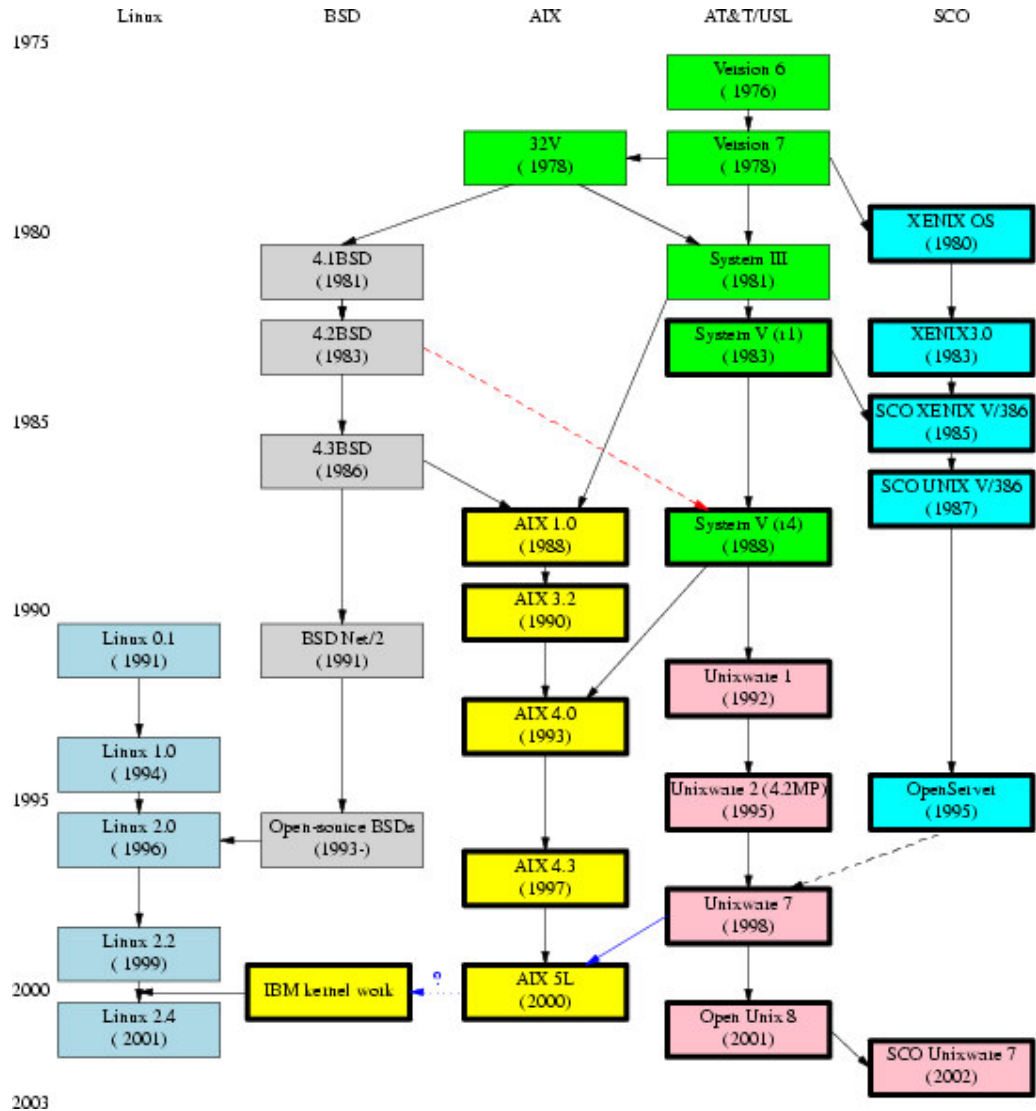


Imagem do [artigo http://www.opensource.org/sco-vs-ibm.html](http://www.opensource.org/sco-vs-ibm.html) - ver legenda

Um pouco antes dessa disputa, em 1991, tendo aprendido sobre sistemas operacionais com sistemas derivados do MULTICS (Ultrix e Minix) na Universidade de Helsinki, Linus Torvalds licenciava, sob GPL (General Public License), o seu kernel Linux. O Linux, por sua vez, encaixou-se na lacuna que restava no projeto de sistema operacional GNU da FSF.

Alguns atribuem o sucesso do GNU/Linux, em detrimento do FreeBSD dentre os sistemas livres, à estagnação e à incerteza que representou o ano de 1994 para os fãs do FreeBSD. Mas os tempos eram outros, já que a liberdade de se buscar conhecimento ainda não ameaçava o que viria a ser o maior negócio da história.

Hoje, encontrando-se em dificuldades econômicas a SCO deseja extorquir, com base em etéreos e duvidosos títulos de propriedade que prefere não tornar públicos, quem venceu a lógica da avareza e o medo de incertezas para investir em software livre. Depois de ameaçá-los com vagas alegações de propriedade sobre "métodos e conceitos" do Unix, por conta de um contrato que admite ser confuso, a empresa veio posteriormente a admitir que as evidências de violação se resumem a semelhanças no código-fonte do seu System V e várias distribuições Linux. Ou seja, trata-se, por enquanto, de uma disputa restrita a copyrights.

Pois então qual código, exatamente, teria sido "pirateado" do seu sistema? Segundo Bill Craybook, professor e programador de kernel de sistemas operacionais, contratado pela SCO para examinar e comparar códigos-fonte do system V e de várias distribuições Linux, o que pôde ele encontrar, num primeiro exame superficial, foram apenas 80 linhas de código semelhantes, em uma função do kernel. O instigante e o sintomático, entretanto, é que a função na qual foi encontrada código replicado parece ser originária do freeBSD.

Confirmada a origem (<http://www.osopinion.com/perl/story/21722.html> - [cached](#)), a entrada desse código no System V teria se dado através do acordo de cooperação entre a AT&T e universidades, legitimada pelo acordo entre Novell e Berkeley, sendo que a licença de sua fonte original -- o freeBSD -- permite o livre reuso, sem qualquer discriminação ou privilégios.

Em sua insistência, os advogados de Berkeley que negociaram o acordo com a Novell queriam exatamente

prevenir o tipo de confusão que a SCO parece estar interessada em explorar. Enquanto isto, a Microsoft resolveu se adiantar, fazendo um acordo com a SCO em valor não declarado, a título de licença de uso de sistemas da família Unix ou demais códigos da empresa, emprestando credibilidade a esta forma de extorsão. Na informática, essa estratégia é conhecida como FUD (Fear, Uncertainty and Doubt).



As tais "patentes de métodos e conceitos do Unix" são como moeda podre no jogo de azar da indústria monopolista de software proprietário. Se existirem, serão como máquinas lotéricas na mão de magistrados e jurados. Os escritórios de patentes, com suas etéreas indulgências ao mundo dos bits, ameaçam cumprir o papel dos cartórios numa nova Inquisição, lançando-nos numa nova Idade Média, como preconizou George Orwell em 1984, se nossa sociedade se deixar enfeitiçar pela religião da avareza. Se a Justiça e os negociadores de acordos internacionais desprezarem os valores humanistas, pela supremacia de uma cenoura mítica que nos balança o fundamentalismo neoliberal.

Nosso maior desafio doravante está no coração e na mente de magistrados e juristas. E os sapos piramidais? São os que se conformam com a supressão gradual das liberdades

humanas, resgatadas a duras penas de negras paginas da história, em nome de um totalitarismo capitalista auto-organizador e seu esquema de pirâmide da prosperidade, promovido pelos devotos da religião da avareza, religião que prega a conversão forçada dos infiéis. O sapo, por ser um batráquio, pode ser cozinhado vivo pelo lento e gradual aquecimento da água fria, já que não sente a temperatura, mas apenas sua mudança abrupta. Como os aztecas e seus prisioneiros, cozinhados lentamente pelos dogmas dessa mesma religião.



### **O sapo piramidal**

v.2                      revisado  
v.3 revisado em 05/07/03

em

14/06/03