

LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO
RIO GRANDE DO NORTE

FORENSE COMPUTACIONAL COMO FERRAMENTA DE APOIO AO ENSINO DA
DISCIPLINA DE SISTEMAS OPERACIONAIS
FORENSE COMPUTACIONAL COMO FERRAMENTA DE APOIO AO ENSINO DA
DISCIPLINA DE SISTEMAS OPERACIONAIS

Monografia apresentada à Coordenação do Curso de
graduação em Licenciatura em Computação da Faculdade
Natalense para o Desenvolvimento do Rio Grande do Norte,
como requisito parcial para obtenção do título de
Licenciado em Computação.

ROBERSON BARBOSA DE CARVALHO

Orientador: Ricardo Kleber Martins Galvão

Natal, Novembro de 2005

Natal, Novembro de 2005

LICENCIATURA EM COMPUTAÇÃO
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO
RIO GRANDE DO NORTE



FORENSE COMPUTACIONAL COMO FERRAMENTA DE APOIO AO ENSINO DA DISCIPLINA DE SISTEMAS OPERACIONAIS

ROBERSON BARBOSA DE CARVALHO

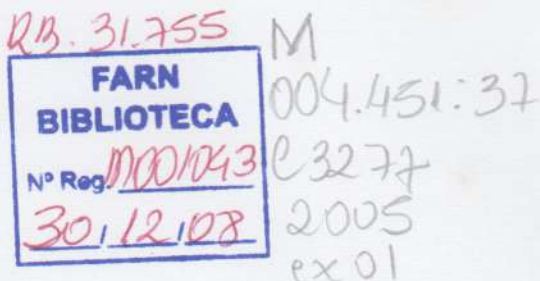
Monografia apresentada à Coordenação do Curso de
graduação em Licenciatura em Computação da Faculdade
Natalense para o Desenvolvimento do Rio Grande do Norte,
como requisito parcial para obtenção do título de
Licenciado em Computação.

Orientador: Ricardo Kléber Martins Galvão

MEMBRO EXAMINADOR

MEMBRO EXAMINADOR

Natal, Novembro de 2005



Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Carvalho, Roberson Barbosa de
Forense computacional como ferramenta de apoio ao ensino da disciplina de
Sistemas Operacionais / Roberson Barbosa de Carvalho. – Natal: FARN, 2005.
56 f.

Orientador: Ricardo Kleber Martins Galvão (MSc.)
Monografia (Graduação em Licenciatura em Computação) – Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte.

1. Sistemas Operacionais – ensino – monografia 2. Forense Computacional –
monografia I. Título

RN/FARN/BC

CDU 004.451:37

LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO
RIO GRANDE DO NORTE

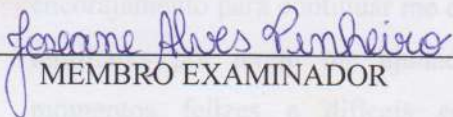
FORENSE COMPUTACIONAL COMO FERRAMENTA DE APOIO AO ENSINO DA
DISCIPLINA DE SISTEMAS OPERACIONAIS

ROBERSON BARBOSA DE CARVALHO

Monografia apresentada e aprovada em _____ de _____ de 2005

ORIENTADOR

MEMBRO EXAMINADOR


MEMBRO EXAMINADOR

Natal, Novembro de 2005

Roberson Barbosa Carvalho

AGRADECIMENTOS

Agradeço primeiramente a Jesus Deus por conceder força e inteligência suficiente para que pudesse conquistar meus objetivos, buscando sempre coragem para enfrentar os desafios propostos pela vida.

Meus queridos pais, irmão, noiva e familiares, pela força dada no decorrer dessa jornada, me apoiando em todos os momentos, sejam eles difíceis ou fáceis, me encorajando a nunca desistir dos meus sonhos.

Ao meu orientador Ricardo Kléber Martins Galvão, por sua dedicação, paciência e confiança que teve com minha pessoa, ajudando a concluir esta monografia, ao meu co-orientador João Tadeu Weck por ter me aconselhado em determinadas etapas de minha conclusão, a todos os professores.

A todos os meus amigos ajudando a conseguir concluir profissional e pessoal.

Aos meus pais Reginaldo e Célia que me incentivaram durante todo o período acadêmico, ensinando que o estudo e o conhecimento formam a base para uma boa honestidade, educação e um ótimo futuro profissional e pessoal. Ao meu irmão Rogério e família, que me incentivam a continuar estudando, à minha noiva Luana e familiares, por terem mais compreensão e me dar mais coragem para continuar a lutar para atingir meus objetivos, a meu tio Melo e tia Elita por terem me hospedado e me dado todo o suporte na sua residência, aos meus avós Sr. Lauro, Sra. Iolanda, Sr Luiz e Sra Maria pelos seus conselhos, aos meus amigos Alexandre, Aldemir, Claudemir, Dickson, Israel, Kayo, Nélio, Paulo, Rodolfo, Rudson, Wander, Wandemberg, Welber e os demais, sempre me estimulando e dando encorajamento para continuar me esforçando. A todos meus familiares por terem me ajudado e compartilhado em momentos felizes e difíceis encontrados em toda a caminhada até a conclusão da minha graduação. Ao meu Orientador Ricardo Galvão, por ter bastante paciência e por ter acreditado no meu trabalho, enfim a todos os professores que acompanharam toda a minha graduação e ter acreditado no meu esforço e dedicação para conseguir terminar minha monografia.

Roberson Barbosa Carvalho

AGRADECIMENTOS

Agradeço primeiramente a Jeová Deus por conceder força e inteligência suficiente para que pudesse conquistar meus objetivos, buscando sempre coragem para enfrentar os desafios propostos pela vida.

Meus queridos pais, irmão, noiva e familiares, pela força dada no decorrer dessa jornada, me apoiando em todos os momentos, sejam eles difíceis ou fáceis, me encorajando a nunca desistir dos meus sonhos.

Ao meu orientador Ricardo Kléber Martins Galvão, por sua dedicação, paciência e confiança que teve com minha pessoa, ajudando a concluir esta monografia, ao meu co-orientador João Tadeu Weck por ter me aconselhado em determinadas trajetórias da minha conclusão, a todos os professores que me acompanharam todo a graduação.

A todos os meus amigos e colegas que me ajudaram, dando seus conselhos e me ajudando a conseguir concluir a minha graduação, por acreditarem no meu potencial profissional e pessoal.

RESUMO

Esta monografia está focada na demonstração da aplicabilidade de uso da forense computacional no ensino da disciplina de sistemas operacionais, tendo em vista que as metodologias geralmente utilizadas pelos docentes desta disciplina possuem uma abordagem essencialmente teórica. Esta demonstração consiste no desenvolvimento de um modelo prático de abordagem da disciplina pelo docente, com o objetivo de fornecer ao discente uma melhor percepção do conhecimento de certas características do funcionamento de um sistema operacional com fundamentação pedagógica. A disciplina de sistemas operacionais faz parte da grade curricular da maioria dos cursos superiores na área da computação. Forense computacional é uma área multidisciplinar, com forte ênfase no uso de ferramentas da computação, que visa a exploração e investigação de recursos computacionais com finalidades de preservação e recuperação de evidências digitais.

Palavras-Chaves: Sistemas de arquivos, Instrucionista e Forense.

ABSTRACT

This monograph possesses a focus in the demonstration of the applicability of forensic use of the computational one in the education of disciplines of operational systems, in view of that the methodologies generally used by the professors of this discipline possess a boarding essentially theoretician. This demonstration consists of the development of a practical model of boarding of disciplines for the professor, with the objective to supply to the learning one better perception of the knowledge of certain characteristics of the functioning of an operational system with pedagogical recital. It disciplines it of operational systems is part of the curricular grating of the majority of the high school in the area of the computation. Forensic computational it is an area to multidiscipline, with strong emphasis in the use of tools of the computation, that aims at the exploration and inquiry of computational resources with preservation purposes and recovery of digital evidences.

Key words: Filesystems, Instrucion e Forensic.

4.2 SISTEMA DE ARQUIVOS	17
Key words: Filesystems, Instrucion e Forensic.	20
4.4 INTERSEÇÃO INFORMÁTICA E EDUCAÇÃO	21
7 DESENVOLVIMENTO	27
7.1 ESTUDO DE CASO Nº 01 – UTILIZANDO O ILS PARA APRESENTAR ESTRUTURA DO SISTEMA DE ARQUIVO EXT2 (UNIX)	28
7.2 ESTUDO DE CASO Nº 02 – UTILIZANDO O UNRM PARA APRESENTAR INFORMAÇÕES CONTIDAS NOS L-NODES	32
7.3 ESTUDO DE CASO Nº 03 – UTILIZANDO O GRAVE-ROBBER COM MACTIME PARA APRESENTAR INFORMAÇÕES DE DATAS DE MODIFICAÇÃO, ACESSO E CRIAÇÃO DOS ARQUIVOS	35
7.4 PROPOSTA DE AULA DE SISTEMAS DE ARQUIVOS UTILIZANDO FERRAMENTAS FORENSE	38
7.5 UTILIZANDO A FERRAMENTA ILS E A UNRM PARA MELHORAR A COMPREENSÃO DO ALUNO NO ASSUNTO IMPLEMENTAÇÕES DE SISTEMAS DE ARQUIVOS	38
7.6 UTILIZANDO A FERRAMENTA GRAVER-ROBBER E A MACTIME PARA MELHORAR A COMPREENSÃO DO ALUNO NO ASSUNTO ATRIBUTOS DE ARQUIVOS	39
8 ANÁLISE DE RESULTADOS	41
9 CONCLUSÃO	42
10 REFERÊNCIAS BIBLIOGRÁFICAS	43
11 BIBLIOGRAFIA CONSULTADA	44
ANEXO A – LICENÇA PÚBLICA GERAL GNU	46
ANEXO B – QUESTIONÁRIO	53

SUMÁRIO

1 INTRODUÇÃO	9
2 FORENSE COMPUTACIONAL	11
3 THE CORONER'S TOOLKIT (TCT)	12
3.1 LAZARUS.	12
3.2 UNRM.	12
3.3 GRAVE-ROBBER.	13
3.4 MACTIME.	13
3.5 ILS E ILS2MAC.	13
3.6 PCAT.	13
4 SISTEMA OPERACIONAL	14
4.1 GERÊNCIA DE PROCESSO.	14
4.2 SISTEMA DE ARQUIVOS.	17
5 LINUX	20
6 A INTERSEÇÃO INFORMÁTICA E EDUCAÇÃO	21
7 DESENVOLVIMENTO	27
7.1 ESTUDO DE CASO Nº 01 – UTILIZANDO O ILS PARA APRESENTAR ESTRUTURA DO SISTEMA DE ARQUIVO EXT2 (UNIX)	28
7.2 ESTUDO DE CASO Nº 02 – UTILIZANDO O UNRM PARA APRESENTAR INFORMAÇÕES CONTIDAS NOS I-NODES.	32
7.3 ESTUDO DE CASO Nº 03 – UTILIZANDO O GRAVE-ROBBER COM MACTIME PARA APRESENTAR INFORMAÇÕES DE DATAS DE MODIFICAÇÃO, ACESSO E CRIAÇÃO DOS ARQUIVOS.	35
7.4 PROPOSTA DE AULA DE SISTEMAS DE ARQUIVOS UTILIZANDO FERRAMENTAS FORENSE.	38
7.5 UTILIZANDO A FERRAMENTA ILS E A UNRM PARA MELHORAR A COMPREENSÃO DO ALUNO NO ASSUNTO IMPLEMENTAÇÕES DE SISTEMAS DE ARQUIVOS.	38
7.6 UTILIZANDO A FERRAMENTA GRAVER-ROBBER E A MACTIME PARA MELHORAR A COMPREENSÃO DO ALUNO NO ASSUNTO ATRIBUTOS DE ARQUIVOS.	39
8 ANÁLISE DE RESULTADOS	41
9 CONCLUSÃO	42
10 REFERÊNCIAS BIBLIOGRÁFICAS	43
11 BIBLIOGRAFIA CONSULTADA	44
ANEXO A – LICENÇA PÚBLICA GERAL GNU	46
ANEXO B – QUESTIONÁRIO	55

Figura 1 – Instalação do tct no Debian GNU/Linux.	27
Figura 2 – Formatação do disquete do tipo de sistemas de arquivos ext2.	27
Figura 3 – Criando a imagem do disquete Bit a Bit.	28
Figura 4 – Usando o aplicativo IIs com argumento -e.	28
Figura 5 – Usando o aplicativo IIs com argumento -a.	30
Figura 6 – Usando o aplicativo IIs com argumento -A.	31
Figura 7 – Usando o aplicativo Unrm com argumentos -e -f ext2fs com i- node 45.	32
Figura 8 – Usando o aplicativo Unrm com argumentos -e -f ext2fs com i- node 46.	33
Figura 9 – Usando o aplicativo Unrm com argumentos -e -f ext2fs com I- node 47.	34
Figura 10 – Montando a imagem do disquete no diretório mntforensic.	35
Figura 11 – Usando grave-robber para uma coleta de informações.	36
Figura 12 – Usando o aplicativo Mactime para apresentar os atributos das datas de criação, modificação e acesso dos arquivos.	37

1 INTRODUÇÃO

Atualmente com o desenvolvimento do mundo globalizado e com as novas tecnologias surgindo de maneira avassaladora, surge a necessidade de que o homem desenvolva mecanismos de proteção contra os problemas que possam surgir com o uso dessa tecnologia. A informática vem sendo inserida no cotidiano dos seres humanos há algum tempo, concentrando atualmente como um dos principais meios de comunicação utilizados pelo homem.

Com o advento da Internet, e com equipamentos que possibilitam a comunicação entre os seres humanos de todas as partes do mundo, com o uso da Internet é possível realizar diversas atividades como a realização de transações bancárias, visitar qualquer parte do mundo interconectada à Internet, envio de correspondências para algum parente ou amigo que esteja em outro território, através de e-mail, utilização do computador como ferramenta de comunicação em tempo real, através de *chats* (sala de bate-papo pela Internet), usando a tecnologia de telefonia pela Internet (voz sobre IP), aplicações de mensagens em tempo real, como o MSN, ICQ, os canais de IRC (*Internet Relay Chat*), dentre outras maneiras de uso da mesma. Tudo isso sendo transmitido eletronicamente, utilizando dispositivos digitais desenvolvidos para tal propósito. Juntamente com todas essas vantagens trazidas pela Internet, veio uma questão bastante importante que é a segurança das informações que por ela trafegam. Existem falhas em programas, vírus, programas espiões (*trojans*, *keyloggers*) e outros softwares maléficos. Programas desta natureza vêm se tornando cada vez mais freqüentes em uma sociedade que utiliza cada vez mais a tecnologia no seu dia-a-dia. Sabe-se que existem riscos reais de algum dia nós todos lidar com essa realidade, pois todos os que estão conectados à Internet estão passíveis da ação destas “pragas”. Segundo Reis (2003), um número crescente de criminosos faz uso de computadores, servidores de rede, equipamentos que estejam conectados à Internet para suas operações ilícitas. Em paralelo a estes problemas trazidos pela Internet, têm surgido diversos tipos de pesquisas para o combate destes criminosos.

Nestes últimos anos a informática vem ocupando um espaço cada vez maior em todas as esferas da sociedade, tornando o seu domínio um requisito quase indispensável aos indivíduos que buscam inserir-se no mercado de trabalho e interagir com a sociedade como um todo.

Tais fatos obrigam docentes e demais gestores de cursos de níveis técnicos, profissionalizantes e superiores a buscarem alternativas que facilitem o aprendizado e garanta maior segurança às informações.

Tendo em vista a disciplina sistemas operacionais – que compõe a grade curricular dos principais cursos na área de computação, em nível superior – e pelo fato de apontar para uma metodologia essencialmente teórica, surge naturalmente a necessidade de pesquisas por meios práticos que relacionem a teoria com a prática. Todo processo de ensino requer métodos que possibilitem ao aluno praticar o material estudado, pois o processo de aprendizado é facilitado pela relação da teoria com a prática.

Forense computacional é um conjunto de práticas digitais investigativas, criado pela necessidade de combater tecnicamente as evoluções dos crimes com utilização de computadores e a Internet, e possui como objetivo estudar meios de isolamento, preservação e recuperação de evidências digitais que venham a servir como provas de autoria de delitos realizados por criminosos.

Na disciplina de sistemas operacionais, têm-se por definição que um sistema operacional cliente/servidor básico é composto por cinco módulos: a gerência de processos, gerência de memória, sistemas de arquivos, gerência de dispositivos de entrada/saída e o kernel. A metodologia comumente utilizada na docência desta disciplina baseia-se em apresentações da parte teórica de todos esses componentes, além do projeto, desenvolvimento e implementação atual nos sistemas operacionais modernos.

Utilizando ferramentas da forense computacional têm-se a possibilidade de mostrar estruturas que compõem o sistema operacional em complemento ao conteúdo teórico ministrado, com isso o aluno terá uma melhor percepção do assunto estudado.

2 FORENSE COMPUTACIONAL

A forense computacional é uma ciência que tem como objetivo estudar e analisar evidências digitais, em sistemas operacionais ou máquinas comprometidas que estejam envolvidas em operações ilegais ou que foi alvo de acesso não autorizado. Tem como função procurar indícios ou provas digitais que incriminem ou inocentem um determinado indivíduo ou grupo de pessoas que estejam envolvidos no caso.

O termo forense está inserido no meio jurídico e policial, onde se tenta investigar crimes, e solucionar mistérios. Envolve a ajuda de peritos criminalistas, da área jurídica, que fazem uso de métodos e ferramentas para analisar cuidadosamente e preservar o material recolhido na cena do crime. Um exemplo bastante comum é quando ocorre algum delito, como um assassinato. Os policiais têm como principal função preservar a cena do crime para que nada possa acontecer com as evidências que reside no local, deixado pelo criminoso, pois caso isso ocorra, pode prejudicar a ação dos peritos, que conseqüentemente pode influenciar nos resultados da análise. Numa investigação envolvendo a Informática, como sendo a arma do crime ou a vítima, os policiais e peritos utilizando meios legais, tem como objetivo recolher qualquer material que possa conter algo bastante importante, e que tenha alguma relevância no processo criminal e na análise dos fatos, como disquetes, algum tipo de documentação, computadores e outras mídias. Todo esse material deve permanecer sobre vigilância e preservação da sua integridade o tempo todo e em local seguro. Nesse contexto, temos como principal elemento o computador, como sendo uma fonte de evidências digitais, em que o perito, utilizando ferramentas e métodos legais possam extrair indícios no seu exame forense, em busca do seu principal objetivo. Segundo Guimarães (2003), atualmente no Brasil não existe uma legislação específica para o cumprimento da investigação forense computacional. Os peritos têm como referência a essa prática documentações, registros e estudos realizados cientificamente, para poderem usar como metodologia de análise. Várias pesquisas vêm sendo realizadas em busca de padronização e metodologias que auxiliem o perito na sua função. O perito realiza suas investigações em computadores, precisamente varrendo alguns módulos importantes que compõem um sistema operacional, como por exemplo, o sistema de arquivo.

3 THE CORONER'S TOOLKIT (TCT)

Este trabalho fará uso de algumas ferramentas da área forense, como foi discutido anteriormente. Utilizaremos softwares livres, como é o caso do The Coroner's Toolkit (TCT). Estas ferramentas livres possuem seu código fonte aberto, sua Licença é GPL (General Public License). Veja o anexo A. O TCT foi desenvolvido por Dan Farmer e Wietse Venema, dois importantes membros da comunidade do software livre. O TCT foi desenvolvido em uma linguagem de baixo nível, em C, e roda no SO Linux e realiza a perícia em sistema de arquivos *ext2* (padrão Unix).

O TCT é composto por utilitários como o *lazarus*, *unrm*, *grave-robber*, *mactime*, *ils*, *ils2mac* e o *pcat*. O objetivo desta monografia é apresentar algumas destas ferramentas no auxílio do ensino da disciplina de sistemas operacionais, de modo que os alunos podem aplicar seus funcionamentos de maneira acadêmica e pedagógica.

3.1 LAZARUS

O *lazarus* é uma das principais ferramentas do TCT, pois a função é varrer toda a área do disco comprometido, dados em memória, área não utilizada do disco a procura de informações que não foram sobrescrito. É bastante utilizada em conjunto com a ferramenta *unrm* na recuperação de arquivos. O resultado do *lazarus* é um conjunto de caracteres pré-codificados, que pode ser apresentado na tela ou armazenado em arquivo com extensão *html* e visualizado por meio de qualquer navegador de Internet.

3.2 UNRM

O *unrm* tem seu principal objetivo explorar toda a tabela de *i-nodes* do sistema de arquivos *ext2*, abrindo blocos de dados que possam existir no disco comprometido. Podendo assim recuperar arquivos em determinadas áreas ou setores da mídia os dados não alocados na tabelas de *i-nodes*. O perito tem como objetivo, utilizando esta ferramenta, fazer uso da mesma para procurar informações que possa servi de provas contra o individuo que está sendo indiciado por crimes digitais. Tais provas são evidências digitais que os peritos de informática usam para culpar ou inocentar o individuo que está sendo julgado.

3.3 GRAVE-ROBBER

Esta é uma ferramenta bastante útil do TCT, pois a principal função é a coleta de informações em sistema de arquivos *ext2* comprometido, como informações de processo, arquivos de log, arquivos de configuração, e outras informações relevantes para o perito.

3.4 MACTIME

O *mactime* é uma ferramenta que utiliza o resultado do *grave-robber* para fazer uma relação das datas de modificação, dos acessos e criação dos arquivos coletados, fazendo uma cronologia do tempo destas datas. Com isso determinar o intervalo de tempo entre a data antes do incidente e determinar quais arquivos foram modificados.

3.5 ILS e ILS2MAC

Essa ferramenta tem função de apresentar a tabela de *i-nodes* de sistemas de arquivos *ext2*, com isso podemos observar com o resultado, quais arquivos estão alocados nas entradas da tabela ou não. Com isso determinar quais possíveis arquivos foram apagados. Já o *ils2mac* gera uma lista com datas de criação, modificação e acesso dos arquivos existente na mídia explorada.

3.6 PCAT

O *pcat* consiste em uma ferramenta, desenvolvida por Wietse Venema para o TCT, que possui uma finalidade de capturar dados de processos na memória RAM em sistemas *Unix*. Exige privilégios de super-usuario do sistema para a execução da ferramenta, e requer que o sistema esteja *on-line* (ligado). O perito só tem que informar o número do processo (*pid*).

4 SISTEMA OPERACIONAL

Será apresentada uma introdução ao Sistema Operacional (SO) e suas partes, descrevendo quais componentes da sua estrutura que vai ser utilizado para a conclusão da pesquisa. Todo SO existente atualmente tem como função controlar os recursos físicos (*Hardware*¹) que faz parte do computador. Controla o *hardware* e as aplicações que os usuários interage. Segundo Tanenbaum (2000), o SO é composto de sistemas (*Softwares*) que gerencia e executam os processos de operação do computador e os programas aplicativos que realizam operações que o usuário deseja. Na estrutura de um SO, existem módulos que fazem parte da sua composição. Tais módulos trabalham em parceria, pois um depende do outro para executarem suas tarefas. O SO é composto por cinco núcleos: o sistema de arquivos, gerência de memória, gerência de processos, sistema de entrada/saída e o *kernel*. *Kernel* é um núcleo onde o SO utiliza para suas operações de baixo nível, executando no modo de supervisor, que é protegido do usuário para não ter acesso direto ao *hardware*. O computador se divide em duas partes. Pode ser dividido em partes físicas que são os componentes como placa-mãe, processador, memória, disco rígido, dentre outros periféricos e circuitos. A parte lógica onde estar envolvido todo tipo de *software* que o computador pode possuir, como exemplo, aplicações do usuário, e o próprio SO. Existe um modelo para representar um computador, segundo Tanenbaum (2000) de modo simplificado, embaixo fica a parte física, logo acima estar o SO, por sua vez existem ligações (*system calls*- chamadas de sistema) que se ligam as aplicações por onde o usuário têm acesso.

O foco da pesquisa tem como objetivo estudar uma maneira em que as ferramentas de forense computacional podem ajudar, de maneira prática, o estudo da disciplina de SO. Por isso serão apresentadas algumas partes do SO envolvido na ciência forense, pois nem todas as ferramentas da forense têm a possibilidade de trabalhar com todos os núcleos do SO. Descreveremos na nossa pesquisa os núcleos de sistema de arquivos, onde a ciência forense e suas ferramentas têm maior ênfase de estudo, e a gerência de processo, onde a análise forense obtém informações relacionadas com cada processo.

4.1 GERÊNCIA DE PROCESSO

Gerência de Processo é um dos principais núcleos de um SO. Como o nome já sugere, é a parte do SO que gerencia cada processo que o computador pode gerar para uma

¹ Hardware é a parte física. Conjuntos de periféricos e dispositivos que compõem o computador.

determinada tarefa. Um processo é um programa em execução, com ele está relacionados um conjunto de informações, associando com qual endereço de memória estar sendo referenciado, qual registrador estar envolvido, contador de processo e suas informações necessárias para sua execução.

Todas essas informações e outras, que cada processo possui, a gerência de processo armazenam em uma tabela dinâmica. Nessa tabela existem campos que identificam cada processo que está sendo executado na memória ou suspenso. A tabela dinâmica utiliza vários campos para determinar cada processo.

É fundamental que cada SO possua na sua tabela dinâmica de processo, os principais campos: PID (Identificador de processo), estado do processo (pode ser executando, pronto ou bloqueado), alocação de memória (o endereço de memória onde o processo estar executando), ponteiro de pilha (indicador de memória utilizada), contador do programa (usado para determinar quantas vezes foi executado), tempo de CPU utilizado (apresenta quanto tempo estar executando), registradores (utilizado para sua execução), dentre outros.

Uma característica bastante interessante, é que dependendo do momento, o processo pode gerar um processo-filho, montando uma árvore de processos. Esse processo-filho, na maioria das vezes precisa trocar informações com outros processos. Essa comunicação é chamada de comunicação inter-processos. Existem algumas questões envolvendo a necessidade das comunicações entre processos, as principais são: a passagem de informações de um processo para outro, essa comunicação pode interferir nas informações trocadas entre processos em atividades críticas, a execução seqüencial de processos relacionando a dependência de informação de um processo-filho com relação ao processo-pai. Alguns processos podem utilizar dispositivos de entrada/saída como disquetes, disco rígido, CD's, e outros, na sua execução.

O SO deve ter controle de acesso a tais dispositivos, para isso utiliza um vetor de interrupção, onde são armazenadas informações relacionadas ao seu acesso. Fazendo com que nenhum processo tenha acesso direto ao *hardware*. Outra característica é o multi-processamento, em alguns sistemas que possui essa característica pode estar executando vários processos ao mesmo tempo, na verdade alternando entre eles em algumas unidades de milissegundos, dando uma falsa ou ilusão de multi-processamento para o usuário, conhecido como pseudoparalelismo. Alguns processos em execução, de acordo com seu nível de

criticidade, pode ter uma variação de tempo constante, em relação a sua execução, são os de tempo real. O SO pode possibilitar a modificação do nível de criticidade, parar e até encerrar cada processo. Por exemplo, um sistema de controle de fluxo aéreo, qualquer variação de tempo nos processos, pode prejudicar bastante o controle dos aviões na hora de quem sobe e desce.

Em alguns SO existem bastantes processos rodando e acessando áreas de armazenamento como dispositivos ou arquivos compartilhados. As condições de corrida (*race conditions*), acontecem constantemente, pois dois ou mais processos têm como tarefa ter acesso e utilizar os mesmos arquivos ou dispositivo compartilhado, tendo permissão para ler ou gravar, principalmente em um endereço de disco que a localização não muda. Cada processo tem um determinado *clock* (tempo) para ser executado.

Imaginamos que um processo "X" tem que terminar a sua execução naquele momento, se por acaso o tempo terminar e ele não pôde ser concluído, pode acontecer que o próximo processo "Y" utilize a mesma área do processo "X" para gravar algo, quando o processo "X" voltar a sua execução, pode retornar um resultado não agradável ao processo que lhe originou. Para evitar as condições de corrida, que mais de um processo utilize uma determinada área compartilhada, deve-se proibir qualquer processo, ao mesmo tempo de ler e gravar nesta área, isso é chamado de exclusão mútua. É uma maneira de impedir que outros processos utilizem uma área compartilhada, que no momento está ocupada por um processo. As partes dos programas em que a memória é compartilhada é acessada, são chamadas de regiões críticas ou sessões críticas.

Com tantos problemas apresentados em relação à condição de corrida e a exclusão mútua como uma prevenção, várias propostas de soluções foram desenvolvidas para tal propósito. Soluções como a desativação das interrupções, variáveis de bloqueio, alternância escrita, a solução de Peterson e a instrução TSL.

Quando o SO se depara com dois processo que estão pronto para ser executado, o próprio tem que decidir quem vai ser processado primeiro. A gerência de processo utiliza alguns algoritmos de escalonamento para fazer um agendamento, com isso organizar e manter uma determinada ordem dos processos a serem executados. Essa tarefa de agendar é realizado por um escalonador. O objetivo do escalonador é fazer com que cada processo seja executado da melhor maneira e no tempo certo, e outra característica é que nenhum processo pode

influenciar a execução de outro. Por isso existem critérios de agendamento, que o próprio tem que ser justo, pois terá que dividir em partes iguais, para todos os processos envolvidos, o tempo de utilização da CPU. Outra meta do escalonador é a eficiência que possibilita o uso do maior nível possível da CPU para o processo em execução. O tempo de resposta também faz parte das metas, pois quanto mais rápido a CPU fornecer o resultado, melhor será para o usuário. Em relação ao tempo de processamento, deve minimizar, pois o usuário deve esperar pouco tempo de saída (*Turnaround*), e o SO deve economizar tempo de processamento, significa executar o maior número de processo naquele tempo disponível (*Throughput*). O que dificulta o SO é que não tem como adivinhar qual processo vai ser executado, pois são diferentes e imprevisíveis. O escalonador faz uso do relógio interno da máquina para sua tarefa. Existe o escalonador preemptivo, que possibilita a suspensão temporária de processos logicamente executáveis, e o não-preemptivo que é utilizado em sistemas *batch*, que o processo é executado até ser concluído.

Quando o SO trabalha em sistemas onde o escalonador é preemptivo, faz uso de algoritmos, como dito anteriormente, para fazer um adequado agendamento de processos. Alguns algoritmos como o de Round Robin, que possui como principal característica, é que cada processo recebe um valor de tempo para ser executado (*quantum*), se ocorrer que o processo não termine naquele *quantum*, imediatamente a condição de “executando” passa para outro processo. O algoritmo de agendamento por prioridade define que cada processo terá um peso, para determinar seu nível de criticidade, o processo pronto para executar estiver com o valor do peso maior será o próximo a ser executado, e toda execução do processo, o valor é decrescido. O algoritmo de Múltiplas Filas já defende uma criação de classes para separar os processos de acordo com seu nível de criticidade, quanto o processo gastar seu tempo e não for suspenso, sobe uma classe. O algoritmo de Menor trabalho primeira será bastante utilizado em sistemas de lotes, pois de antemão já terá um determinado tempo estimado para cada processo, isso diminui a média do *turnaround*. O algoritmo Garantido é que o SO verifica quantos usuários estão utilizando a máquina, estão n logados, e divide em um momento em $1/n$ de processamento da CPU para cada um.

4.2 SISTEMA DE ARQUIVOS

Todo usuário que utiliza um computador tem por motivos armazenar arquivos ou guardar qualquer informação e depois poder recuperá-lo. Os processos disparados pelo computador podem utilizar um determinado arquivo para ler ou gravar informação. O SO tem

um núcleo chamado de Sistema de arquivo que implementa essa função. A ciência forense estuda meios de investigação em máquinas comprometidas, tendo como objetivo varrer toda a máquina a procura de evidências digitais. Essa investigação é realizada em sistema de arquivos, por isso o próprio é bastante importante neste trabalho, pois a maioria das ferramentas forense que vão ser utilizadas para a fundamentação do trabalho, terá uso em cima de um sistema de arquivos. Serão abordados conceitos e como um sistema de arquivos é implementado, com suas principais características.

O sistema de arquivos tem por principal característica definir um nome de arquivo que será armazenado. Um arquivo é um mecanismo de abstração, pois oferece maneiras de armazenamento e leitura. Essa função de armazenar e recuperar são realizados sem que o usuário veja o que realmente está acontecendo. Por isso o sistema de arquivos tem como principal característica a nomeação dos seus objetos de abstração. Em alguns SO, essa característica pode ser implementada de formas diferentes, mais varia em um a oito letras de nome de arquivos válidos. Em outros SO podem existir diferenças em letras maiúsculas e minúsculas e aceitar até 255 caracteres. Em outros casos, onde, um nome de arquivo deve ser dividido em duas partes, por exemplo, um arquivo com o nome "texto.txt", o primeiro nome antes do ponto, define o nome que foi colocado pelo usuário, depois do ponto define qual tipo de arquivo ele é.

Existem estruturas de arquivos, como é implementado e armazenados em disco ou em mídias, que pode ter três possibilidades, a primeira é que um arquivo pode ser estruturado em uma sequência de bytes, que o SO possui função de só armazenar, desconhece o conteúdo do arquivo, pois só enxerga em bytes. A segunda estruturação é de forma de sequência de registros, possuindo comprimento fixo do seu tamanho. A terceira estrutura está relacionada à árvore, possuindo registros de arquivos todos ligados através de campos-chaves em uma posição fixa no próprio. Segundo Tanenbaum (2000), esse modelo de estrutura é bastante utilizado em *mainframes* de grande porte, para seu processamento comercial. Os tipos de arquivos, em que o SO utiliza, é composto por arquivos comuns, diretórios, arquivos especiais de caractere e arquivos especiais de bloco. Suponha que o usuário está digitando um texto em editor de texto e, depois ele salva, esse tipo de arquivo é considerado comum. Diretório é um tipo de arquivo, pois possui a função hierarquicamente estruturar e de manter o sistema de arquivos sempre organizado. Arquivos especiais de Caractere estar relacionado com alguns dispositivos de Entrada/Saída, que pode ser impressoras seriais e outros *hardwares* da mesma categoria. Já os arquivos especiais de bloco, está relacionado com o modo de gravação em

disco, por isso os arquivos são gravados no disco em forma de blocos. Em relação ao acesso dos arquivos, de acordo com os SO atualmente implementa o acesso aleatório. Acesso aleatório é bastante importante para diversos aplicativos que trabalham com rotinas de consulta dos arquivos. Considerando que um arquivo não é formado apenas pelas informações do usuário, mais também por atributos que determinam características de cada arquivo. Esses atributos são campos que são armazenados junto com os arquivos, que podem ter alguns bits reservados para data/hora da criação, a última modificação, o último acesso, o tamanho, o *uid* do dono (identificação do seu criador), a localização no disco, dentre outros campos que podem existir.

O SO utiliza algumas chamadas de Sistemas para gravação e leitura dos arquivos. Chamadas de sistemas mais comuns de um SO: CREATE está relacionada com a criação do arquivo, DELETE está relacionado com a remoção do arquivo, OPEN relaciona com a abertura, CLOSE relaciona com o fechamento do arquivo, entre outras chamadas. O conceito de Diretórios, como uma estrutura do sistema de arquivo para obter uma organização e manter uma hierarquia, é bastante importante, pois o SO quando procurar um arquivo, ele varre seu diretório e então extrai seus atributos e realiza sua operação. Essa organização de hierarquia resulta em um conceito de árvore de diretórios. O sistema de arquivos tem dois tipos de classificação em relação a nomes de caminhos dos seus arquivos, pois é necessário para ajudar o SO quando precisar procurar por eles. Existe um nome de caminho absoluto do arquivo, que consiste deste o diretório raiz, passando por seus subdiretórios (se for o caso) até encontrá-lo. Outro nome de caminho é Relativo, consiste que o arquivo procurado está no diretório atual, onde o usuário está trabalhando.

Existem implementações do sistema de arquivos que podem variar de SO para SO, podendo ser utilizados tipos como alocação contínua, alocação por lista encadeada, alocação por lista encadeada utilizando índices ou nós-i (tabela de *i-nodes*). Nesta última implementação, é utilizada uma tabela que associa com cada arquivo seus atributos e seus endereços de blocos no disco. Quando o SO é iniciado essa tabela é carregada na memória. O SO faz referência entre os arquivos e os endereços de cada bloco na tabela, utilizando ponteiros. Segundo Tanenbaum (2000), quando um arquivo é aberto, seu endereço na tabela de *i-nodes* na memória é localizado, e conseqüentemente é carregado para a memória, onde fica no mesmo até ser fechado.

5 LINUX

O Linux é um SO desenvolvido em 1991 por um estudante da Universidade de Helsinki, na Finlândia, cujo nome é Linus Torvalds. O SO Linux possui código fonte aberto e é liberado para qualquer pessoa ou entidade que queira utilizá-lo, pois é licenciado pelos termos da Licença GPL (*General Public License*), significa que o usuário não precisa pagar para acessá-lo, podendo também copiar e distribuir para todo mundo. Também é disponível para *download*, em determinados *sites* que têm seu principal foco a disseminação do Linux. Como dito anteriormente seu código fonte é aberto, dando possibilidade que o usuário possa olhar suas funcionalidades e, se possível encontrar erros e corrigi-los, essa é uma das possibilidades que o Linux possui e também é um dos motivos no qual obteve muitos adeptos.

Possui diversas características, por exemplo, é Multitarefa, Multi-usuário, podendo ser instalado no mesmo computador tendo outro SO (DOS, Windows, FreeBSD, etc), possui diversos suportes a dispositivos físicos, como placa-mãe, placa de vídeo, placa de rede, dentre outros periféricos que compõe um computador, como suportes a dispositivos lógicos como arquivos e diretórios, sendo seu disco formatados por outros SO, como Novell, OS/2, Windows (FAT, NTFS), etc.

O sistema de arquivos *ext2* é um de disco de alta performance usado nativamente pelo Linux para dispositivos de armazenamento, como discos rígidos e mídias removíveis, por exemplo, unidade de disquete. Esse núcleo que compõe um SO, será trabalhado ao longo desta monografia, pois as ferramentas da forense computacional que será utilizada têm como plataforma de uso e perícia esse tipo de sistema de arquivos.

O sistema de arquivos *second extended Filesystem* (Sistema de arquivo de Segunda Extensão) foi desenhado e projetado como uma extensão de *extended Filesystem* (Extensão de Sistema de arquivos). O *ext2* oferece a melhor performance (em termos de velocidade e uso da CPU) entre todos os sistemas de arquivos suportados pelo Linux.

Ext2 é um sistema de arquivos muito rápido pelo fato de não possuir um *journaling* (ou *log*), significa que todas as mudanças realizadas no sistema de arquivo são armazenadas em *log* (registros), para uma posterior recuperação em caso de falha do disco. O *ext2* grava dados diretamente no disco, se tornando mais rápido. Quando ocorre algum defeito no disco, o aplicativo *fsck* (nativo do Linux) é acionado para a verificação do sistema, corrigindo eventuais perdas de dados.

6 A INTERSEÇÃO INFORMÁTICA E EDUCAÇÃO

A utilização da informática no ambiente escolar tem suas raízes no ensino através das máquinas. Esta idéia foi usada por Dr. Sidney Pressey em 1924 que inventou uma máquina para corrigir testes de múltipla escolha. Já no início de 1950, B.F. Skinner propôs uma máquina para ensinar usando o conceito de instrução programada. Tal modalidade consiste em dividir o material a ser ensinado em pequenos segmentos logicamente encadeados e denominados módulos. Cada fato ou conceito é apresentado em módulos sequenciais, que terminam com uma questão que o aluno deve responder preenchendo espaços em branco ou escolhendo a resposta certa entre diversas alternativas apresentadas. O estudante deve ler o fato ou conceito e é imediatamente questionado. Se a resposta está correta o aluno pode passar para o próximo módulo. Se a resposta é errada, a resposta certa pode ser fornecida pelo programa ou, o aluno é convidado a rever módulos anteriores ou, ainda, a realizar outros módulos, cujo objetivo é remediar o processo de ensino.

No início dos anos 60 diversos programas de instrução programada foram implementados no computador — nascia a instrução auxiliada por computador ou "computer-aided instruction", também conhecida como CAI. Na versão brasileira estes programas são conhecidos como PEC (Programas Educacionais por Computador). A disseminação do CAI nas escolas somente aconteceu com os microcomputadores, gerando uma enorme produção de cursos e uma diversificação de tipos de CAI, como tutoriais, programas de demonstração, exercício-e-prática, avaliação do aprendizado, jogos educacionais e simulação.

Segundo Valente (1999), a modalidade que utiliza o computador como máquina de ensinar pode ser caracterizada como uma versão computadorizada dos métodos tradicionais de ensino. As categorias mais comuns desta modalidade são os tutoriais, exercício-e-prática ("drill-and-practice"), jogos e simulação. Os programas tutoriais constituem uma versão computacional da instrução programada. A vantagem está no fato de o computador poder apresentar o material com outras características que não são permitidas no papel como: animação, som e a manutenção do controle da performance do aprendiz. Mas, os sistemas tutoriais inteligentes apresentam problemas como:

- ✓ A intervenção do sistema no processo de aprendizagem é muito superficial. Ainda é muito difícil implementar na máquina um "bom docente".
- ✓ O tamanho dos programas e recursos computacionais que eles requerem é muito grande e os computadores pessoais não são ainda tão poderosos para permitirem que estes programas cheguem até às escolas.

Já os Programas de exercício-e-prática em geral são utilizados para revisar material visto em classe principalmente, material que envolve memorização e repetição, como aritmética e vocabulário. A vantagem deste tipo de programa é o fato do docente dispor de uma infinidade de exercícios que o aprendiz pode resolver de acordo com o seu grau de conhecimento e interesse. Se o software, além de apresentar o exercício, coletar as respostas de modo a verificar a performance do aprendiz, então o docente terá à sua disposição um dado importante sobre como o material visto em classe está sendo absorvido.

Existem também os Jogos Educacionais cuja abordagem é a de exploração autodirigida ao invés da instrução explícita e direta. Semelhante a estes, são os softwares de Simulação, que envolvem a criação de modelos dinâmicos e simplificados do mundo real. Estes modelos permitem a exploração de situações fictícias, de situações com risco, como manipulação de substância química ou objetos perigosos; de experimentos que são muito complicados, caros ou que levam muito tempo para se processarem, como crescimento de plantas; e de situações impossíveis de serem obtidas, como um desastre ecológico. A Simulação oferece a possibilidade do aluno desenvolver hipóteses, testá-las, analisar resultados e refinar os conceitos. Esta modalidade de uso do computador na educação é muito útil para trabalho em grupo, principalmente os programas que envolvem decisões. Os diferentes grupos podem testar diferentes hipóteses, e assim, ter um contato mais "real" com os conceitos envolvidos no problema em estudo. Nos casos onde o programa permite um maior grau de intervenção do aluno no processo sendo simulado (por exemplo, definindo as leis de movimento dos objetos da simulação) o computador passa a ser usado mais como ferramenta do que como máquina de ensinar.

Quando a abordagem for aquela que utiliza o computador como ferramenta, utiliza-se de programas de processamento de texto, planilhas, manipulação de banco de dados, construção e transformação de gráficos, sistemas de autoria, calculadores numéricos, que são aplicativos extremamente úteis tanto ao aluno quanto ao docente, possibilitando a resolução de problemas através do computador, cujo objetivo é propiciar um ambiente de aprendizado baseado na resolução de problemas. Tal abordagem não é nova e já tem sido amplamente explorada através dos meios tradicionais de ensino. O computador adiciona uma nova dimensão, centrada no fato do aprendiz ter que expressar a resolução do problema segundo uma linguagem de programação.

As linguagens para representação da solução do problema podem, em princípio, ser qualquer linguagem de computação, como o BASIC, o Pascal, ou o Logo. No entanto, deve ser notado que o objetivo não é ensinar programação de computadores e sim como representar

a solução de um problema segundo uma linguagem computacional. O produto final pode ser o mesmo — obtenção de um programa de computador — os meios são diferentes. Assim, como meio de representação, o processo de aquisição da linguagem de computação deve ser a mais transparente e a menos problemática possível. Ela é um veículo para expressão de uma idéia e não o objeto de estudo.

Computador como comunicador auxilia os portadores de deficiência física que não dispõem de coordenação motora suficiente para comandar o teclado do computador podem usá-lo, através de dispositivos especialmente projetados, para captar os movimentos que ainda podem ser reproduzidos, como movimento da cabeça, dos lábios, da pálpebra dos olhos, e com isto permitir que estas pessoas transmitam um sinal para o computador. Este sinal pode ser interpretado por um programa e assumir um significado, uma informação, que levará o computador a executar algo, como usar um processador de texto, um controlador de objetos etc., até mesmo para "falar". Os dispositivos para receber ou emitir um sinal para o computador podem ser os mais variados: desde um simples interruptor até um leitor óptico ou de relevo; ou ainda um sintetizador de voz. A combinação destes dispositivos tem permitido que a escrita convencional seja convertida em Braille ou em algo falado, ou que uma mensagem falada seja impressa em Braille.

Existem duas grandes correntes sobre o uso do computador no ambiente educacional que são a Instrucionista e a Construcionista.

A primeira possui como principais características:

- ✓ Não deixa explícito o pensamento do aluno;
- ✓ Na sua maioria, despertam apenas momentaneamente a motivação;
- ✓ Docente é o responsável por provocar reflexões dos alunos;
- ✓ Necessidade da presença de um instrutor nas aulas.

Já o Construcionismo, teoria desenvolvida por Papert, a partir das idéias de Piaget, Vigotsky e Paulo Freire, se propõe a:

- ✓ Construir o conhecimento progressivamente por meio de ações e coordenações de ações que são interiorizadas e se transformam. Esta experiência que propicia a construção do conhecimento não se caracteriza simplesmente por um fazer ou mesmo por um saber fazer, mas sim por uma reflexão sobre o saber fazer;
- ✓ Aquisição do saber como fruto à reconstrução da atividade humana, a partir da reflexão sobre a experiência, continuamente repensada e reconstruída;
- ✓ Ação (a experiência sobre um objeto físico), Testagem (a reflexão que permite encontrar outros elementos ou objetos para testar hipóteses), Depuração (comparação

dos resultados obtidos com os resultados esperados, visando depurar as idéias) e Generalização (observação de novas experiências com o objetivo de transferir os resultados a outras situações);

- ✓ O homem desenvolve relações entre ação e reflexão por meio da experiência concreta, ou seja, o aluno deixa de ser consumidor de informações quando atua como criador de conhecimento e desenvolve criticamente sua “alfabetização” com o uso de ferramentas informáticas.

Portanto, pode-se assinalar como principais diferença aspectos como:

- ✓ As interações dos programas Instrucionistas enfatizam o software e o hardware (a máquina), com vistas à “ensinar” o aluno sem provocar conflitos cognitivos, já o Construcionismo centra-se no pensamento e na criação, no desafio, no conflito e na descoberta;
- ✓ A abordagem Construcionista tem como premissa principal o fato de que o conhecimento que está sendo trabalhado tenha sentido para o aluno, ou seja, que a aprendizagem seja sintônica.
- ✓ Emprego do conceito de Depuração no sentido de que o aluno procura compreender a sua representação do programa para identificar e corrigir os erros. O erro é considerado “benéfico” porque nos leva a estudar o que aconteceu de errado, e, através do entendimento, a corrigi-lo. Independente da linha a ser optada (Instrucionista ou Construcionista), sob qualquer uma delas, os elementos básicos envolvidos nas atividades são o docente, o aluno, o computador e o software ou programa computacional.

O ambiente informático que integra simultaneamente diversas mídias propicia o desenvolvimento de experiências interativas, mas partem dos mesmos pressupostos comportamentalistas, seja utilizando softwares (há inúmeros disponíveis no mercado), seja utilizando o Ambiente Logo (linguagem de programação).

Ao usar linguagem de programação para representar no computador a forma humana de buscar a solução de um problema, isto é, ao descrever todos os passos que podem levar à solução, seqüência lógica de ações, esta “ensinando” o computador resolver o problema através de um programa.

A cooperação entre saberes, artes e técnicas, por um lado, e a tecnologia, por outro, representa um novo momento na educação, pois a inserção de tecnologias informativas e comunicativas possibilitam ao alunado em geral uma modificação no papel desempenhado até agora. De “consumidores” de informações (veiculadas pelo livro didático e pelo docente), o aluno torna-se “produtor” do seu conhecimento, a partir da elaboração e produção de

materiais que associam o conhecimento científico escolar e o seu conhecimento de vivência no mundo que o cerca.

Paulo Freire (1993) assinala que a entrada do computador no meio escolar deveria começar com o desvendar do conhecimento sobre a presença dos computadores e suas redes, no sentido figurado e técnico, na prática educativa, em todas as suas dimensões. Analisando a articulação entre o saber, as novas tecnologias e educação ao poder afirmou:

Exatamente porque somos programados, somos capazes de pôr-nos diante da programação e pensar sobre ela, indagar e até desviá-la (...) Somos capazes de inferir até na programação da que somos resultado (...) A vocação humana é a de "saber" o mundo através da linguagem que fomos capazes de inventar socialmente (...) nos tornamos capazes de desnudar o mundo e de "falar" o mundo. Só podemos falar do mundo porque transformamos o mundo, e o processo não poderia ser ao inverso. Neste sentido, a linguagem não só é veículo do saber, senão que é saber. Não se pode compreender a vida histórica, social e política dos homens fora dele e da necessidade de saber. É um processo que acompanha a vida individual e social das pessoas no mundo com sua politicidade. Isto tem a ver com a forma de "estar sendo" no mundo; o saber fundamental continua constituindo a capacidade de desvendar a razão de ser do mundo, e este é um saber que nem é superior nem inferior aos outros saberes, senão que é um saber que elucida, que desoculta, ao lado da formação tecnológica (...) É o "saber político" que a gente tem que criar, cavar, construir, produzir para que a pós-modernidade democrática, a pós-modernidade progressista se instale e se instaure contra a força e o poder de uma outra pós-modernidade que é reacionária (...) Necessita-se de homens, de mulheres, que ao lado dos saberes técnicos e científicos, estejam também inclinados a conhecer o mundo de outra forma, através de tipos de saberes não preestabelecidos. A negação disto seria repetir o processo hegemônico das classes dominantes, que sempre determinaram o que podem e devem saber as classes dominadas.

(FREIRE, 1997, p. 78)

A concepção de ensino de informática deve aproximar-se do Paradigma Interacionista, pois este trabalha com as várias dimensões da pessoa humana, tais como: a dimensão física e estética, cognitiva, comunitária e social, ético-valorativa e transcendente, na perspectiva de que educar é promover o desenvolvimento integral da pessoa. Portanto, o computador servirá como uma ferramenta, cabendo ao docente o papel de escolher as melhores alternativas para mediar este desenvolvimento.

A partir disso, percebe-se que o docente deve ser um sujeito ativo, estimulador, renovador e transformador da aprendizagem, necessitando ter uma personalidade de educador, indispensável para bem se relacionar com as crianças e jovens e conceitos fundamentais sobre

a educação, na perspectiva de promover relações humanas, significativas e profundas no ambiente escolar, e devem ser capazes de integrar, a partir das disciplinas que ministram, a cultura escolar e a cultura que cada indivíduo trás do seu meio social.

didática de SO com as ferramentas da Forense Computacional, através de estudos de caso. Nestes estudos de caso são exploradas algumas características do uso das ferramentas, onde o foco principal é apresentar uma proposta para a prática do assunto envolvido.

Sistema de arquivos é um dos componentes mais importantes do SO, e também bastante importante para análise forense, pois nele reside uma grande concentração de evidências. Neste trabalho apresentam-se resultados de ferramentas de manipulação de sistema de arquivos aplicadas à prática do ensino do SO. Utilizam-se algumas das ferramentas do conjunto do TCT, como foi abordado anteriormente, mostrando com detalhes a estrutura de sistema de arquivos *ext2* (padrão *Linux*) e suas características, que podem ser utilizadas tanto apoio ao ensino-aprendizagem dos alunos.

O docente pode utilizar, como uma proposta de prática da disciplina de SO, o *it3* (descrito anteriormente), para apresentar aos alunos a estrutura de um sistema de arquivos, utilizando o SO Debian GNU/Linux como plataforma do experimento.

A instalação foi realizada através do comando:

```
# apt-get install ext
```

Figura 1 – Instalação do *ext* no Debian GNU/Linux.

Todos os comandos realizados a partir de agora foram executados utilizando uma imagem de disquete formada no sistema de arquivos *ext2*. Para a formatação do disquete no tipo *ext2*, foi utilizado o aplicativo *mkfs_ext2*, e para criar a imagem *hfs* a *hfs* foi utilizado o aplicativo *mkfs_hfs*, todos estes aplicativos são instalados por padrão no SO Debian GNU/Linux.

Para a formatação foi utilizado esse comando:

```
# mkfs_ext2 /dev/fd0
```

Figura 2 – Formatação do disquete do tipo de sistemas de arquivos *ext2*.

7 DESENVOLVIMENTO

Neste capítulo são apresentadas algumas propostas de uso da didática de SO com as ferramentas da Forense Computacional, através de estudos de caso. Nestes estudos de caso são exploradas algumas características do uso das ferramentas, onde o foco principal é apresentar uma proposta para a prática do assunto envolvido.

Sistema de arquivos é um dos componentes mais importantes do SO, e também bastante importante para análise forense, pois nele reside uma grande concentração de evidências. Neste trabalho apresentam-se resultados de ferramentas de manipulação de sistema de arquivos aplicados à prática do ensino do SO. Utilizam-se algumas das ferramentas do conjunto do TCT, como foi abordado anteriormente, mostrando com detalhes a estrutura do sistema de arquivos *ext2* (padrão *Unix*) e suas características, que podem ser utilizadas como apoio ao ensino-aprendizagem dos alunos.

O docente pode utilizar, como uma proposta da prática da disciplina de SO, o *ils* (descrito anteriormente), para apresentar aos alunos a estrutura de um sistema de arquivos, utilizando o SO Debian GNU/Linux como plataforma do experimento.

A instalação foi realizada através do comando:

```
# apt-get install tct
```

Figura 1 – Instalação do tct no Debian GNU/Linux.

Todos os comandos realizados a partir de agora foram executados utilizando uma imagem de disquete formatado no sistema de arquivos *ext2*. Para a formatação do disquete no tipo *ext2*, foi utilizado o aplicativo *mkfs.ext2*, e para criar a imagem bit a bit foi utilizado o aplicativo *dd*, todos esses aplicativos são instalados por padrão no SO Debian GNU/Linux.

Para a formatação foi utilizado esse comando:

```
# mkfs.ext2 /dev/fd0
```

Figura 2 – Formatação do disquete do tipo de sistemas de arquivos *ext2*.

O argumento `/dev/fd0` está indicando o dispositivo que deve ser formatado, como o SO Debian GNU/Linux utiliza arquivos do tipo *Links* para dispositivos (*hardware*), então o `fd0` está indicando o *floppy* (disquete).

E para criação da imagem:

```
# dd if=/dev/fd0 of=ext2disk.img
```

Figura 3 – Criando a imagem do disquete Bit a Bit.

O aplicativo `dd` precisa de pelo menos dois argumentos, o primeiro e a informação de origem, e a segunda informação para onde deve criar o arquivo de imagem da informação de origem, em resumo precisa da informação de origem e de destino.

7.1 ESTUDO DE CASO Nº 01 – UTILIZANDO O ILS PARA APRESENTAR A ESTRUTURA DO SISTEMA DE ARQUIVOS EXT2 (UNIX)

Utilizando a ferramenta `ils` pode-se apresentar de modo detalhado a estrutura de um

```
# ils -e ext2disk.img
class|host|device|start_time
ils|n3w5|ext2disk.img|1124354333
st_ino|st_alloc|st_uid|st_gid|st_mtime|st_atime|st_ctime|st_dtime|st_mod
e|st_nlink|st_size|st_block0|st_block1
11|a|0|0|1124275396|1124275549|1124275396|0|40700|2|1024|29|0
12|a|0|0|1124275440|1124275439|1124275440|0|100644|1|11718|30|31
13|a|0|0|1124275440|1124275440|1124275440|0|100644|1|36861|42|43
14|a|0|0|1124275485|1124275485|1124275485|0|100755|1|15150|1025|1026
15|a|0|0|1124275485|1124275485|1124275485|0|100755|1|11080|1041|1042
16|a|0|0|1124275485|1124275485|1124275485|0|100755|1|30475|1052|1053
17|f|0|0|1124275485|1124275485|1124275531|1124275531|100755|0|28017|1083
|1084
18|f|0|0|1124275485|1124275485|1124275531|1124275531|100755|0|43372|1112
|1113
19|f|0|0|1124275485|1124275485|1124275531|1124275531|100755|0|13589|1156
|1157
20|f|0|0|1124275485|1124275485|1124275531|1124275531|100755|0|80033|1171
|1172
21|a|0|0|1124275485|1124275485|1124275485|0|100755|1|143512|1251|1252
22|a|0|0|1124275485|1124275485|1124275485|0|100755|1|94554|1393|1394
23|a|0|0|1124275485|1124275485|1124275485|0|100755|1|22263|126|127
24|a|0|0|1124275485|1124275485|1124275485|0|100755|1|21399|149|150
25|a|0|0|1124275485|1124275485|1124275485|0|100755|1|28368|171|172
26|a|0|0|1124275485|1124275485|1124275485|0|100755|1|14035|200|201
27|a|0|0|1124275485|1124275485|1124275485|0|100755|1|60419|215|216
28|a|0|0|1124275485|1124275485|1124275485|0|100755|1|174183|276|277
29|a|0|0|1124275485|1124275485|1124275485|0|100755|1|60625|448|449
30|a|0|0|1124275485|1124275485|1124275485|0|100755|1|16757|509|510
31|a|0|0|1124275485|1124275485|1124275485|0|100755|1|16353|527|528
32|a|0|0|1124275485|1124275485|1124275485|0|100755|1|215721|544|545
33|a|0|0|1124275485|1124275485|1124275485|0|100755|1|45583|756|757
34|a|0|0|1124275485|1124275485|1124275485|0|100755|1|10888|802|803
35|a|0|0|1124275485|1124275485|1124275485|0|100755|1|67293|813|814
```

Figura 4 – Usando o aplicativo `ils` com argumento `-e`

sistema de arquivos do tipo *ext2*. O docente pode utilizar essa prática quando estiver explicando o assunto sistemas de arquivos, e com isso, pode apresentar que todos os arquivos contêm atributos, e relatar como é implementado, pelo SO, a gravação dos arquivos neste tipo de sistema de arquivos. Na figura 04 foi utilizado o *ils* com o argumento *-e* que indica para a ferramenta que apresente toda a tabela de *i-node* da imagem do disquete, com todas as informações das áreas alocadas e não-alocadas.

Observando o uso do *ils*, na figura 04, o docente pode mostrar com clareza os blocos alocados, indicado pelo campo *st_alloc*, localizado na terceira (3ª) linha do resultado da ferramenta, que pode assumir a letra “a” indicando que o bloco está alocado, ou a letra “f” indicando que está livre, ou “vazio”. Outro ponto que pode ser explorado, pelo docente, é a descrição de alguns desses atributos que é gravado junto com os arquivos. Por exemplo, o campo *st_uid* e *st_gid* guarda a informações do dono e o grupo que pertence esse arquivo, respectivamente. Os campos *st_mtime*, *st_atime* e *st_ctime* informam o tempo em segundos da modificação, acesso e a criação do arquivo, respectivamente. A implementação do sistema de arquivos utilizado para essa pesquisa, utiliza um método de gravação com ponteiros, significa que quando um arquivo é maior, e não pode ser armazenado em uma única entrada da tabela de *i-node*, o próprio é segmentado, e alocado em outra entrada na tabela. No primeiro *i-node* alocado existe um campo que vai identificar quantas ligações foram necessárias para armazenar o arquivo. Pode-se observar esse campo na figura 04, que é *st_nlink*.

7.2 ESTUDO DE CASO Nº 02 – UTILIZANDO O UNRM PARA APRESENTAR INFORMAÇÕES CONTIDAS NOS I-NODES.

Como foi dito anteriormente, este estudo de caso vai apresentar uma ferramenta já descrita anteriormente, que tem como objetivo apresentar informações que compõem a tabela de *i-nodes* do sistema de arquivos *ext2*. Com o uso desta ferramenta o docente pode abordar qual é a implementação e mostrar que cada *i-node* possui um tamanho específico. Também pode ser apresentada a divisão de um arquivo em diversos *i-nodes*. São Utilizados exemplos do estudo de caso anterior.

Observando o resultado do *ils* na figura 06 (estudo de caso nº 01), toma-se as informações como base para esse estudo de caso com a ferramenta *unrm* para apresentar as informações de algumas entradas de *i-nodes* na tabela.

Na figura 07, foram utilizados alguns argumentos passados para a ferramenta *unrm*, o primeiro é o *-e* tem como objetivo solicitar a exibição completa, bit a bit, da informação do *i-node* desejado. No segundo argumento o *-f ext2fs* indica qual tipo de sistema explorado (neste caso o *ext2*), que é a imagem do disquete, e por último é passado o número de *i-node* desejado que é 45. Então a ferramenta exibe todo o conteúdo do *i-node*. Pode-se observar que na figura 06, são apresentados os *i-nodes* não-alocados, mas utilizando o *unrm* pode ser mostrada a informação que está armazenada nesse *i-node*.

```
# unrm -e -f ext2fs ext2disk.img 45

>Samba - Servidor NetBIOS para Unix</h3>

<p align="justify">
  versão 1.0 - 09/10/1999
</p><p align="justify">
  Elvis Pfützenreuter - info@rootshell.com.br

</p><p align="justify">
  O objetivo deste documento é cobrir algumas lacunas da documentação
  "oficial" do Samba. v.g. SMB-HOWTO e manual on-line do arquivo
  /etc/smb.conf. O diretório de documentos do Samba contém diversos
  textos esparsos que lançam luz sobre os detalhes de funcionamento do
  protocolo NetBIOS, e este texto procura trazer as informações mais
  importantes ali contidas.

</p><p align="justify">
  O Samba é um programa complexo e já bastante maduro. Em sua versão
  estável atual - 2.0 - a única coisa que o Samba não suporta de
```

Figura 7 – Usando o aplicativo Unrm com argumentos *-e -f ext2fs* com *i-node* 45.

O docente pode utilizar essa abordagem para explicar como são implementadas a Leitura/Gravação do sistema de arquivos *ext2*, que é utilizado nessa pesquisa, com isso

relacionando teoria com a prática. Informando todo tipo de características deste tipo de sistema arquivos.

Como podemos observar, na figura 08, a ferramenta recebe o *i-node* seguinte que mostra a continuação da informação apresentada na figura 07. Pois esta informação tem características de ser um arquivo de *html* (por causa das *tags*).

```
# unrm -e -f ext2fs ext2disk.img 46

forma
  irrestrita é o conceito de domínios. Especificamente, ele não pode
ser
  PDC ou BDC em redes Windows NT, nem suporta o conceito de servidores
  WINS redundantes. O suporte completo a domínios está previsto para a
  versão 2.1, que não tardará a ser liberada.

</p><h3>
</h3><h3>0. Preâmbulo</h3>

<p align="justify">
  Parece que boa parte da documentação disponível para o Samba
pressupõe
  conhecimento razoável sobre o protocolo NetBIOS. Infelizmente, não é
  bem essa a realidade. Depois de passar alguns apuros e ficar
  dependendo da sorte para que os servidores Linux aparecessem na
janela
  "Ambiente de Rede", criei coragem e imprimi todos os textos da
  documentação esparsa do Samba (/usr/doc/samba*/docs/textdocs/).
Ali,
  finalmente encontrei aquele conhecimento de baixo nível que dava
  remédio a meus azares.

</p><p align="justify">
  O protocolo NetBios foi conc
```

Figura 8 – Usando o aplicativo Unrm com argumentos `-e -f ext2fs` com *i-node* 46.

Na figura 09, é a continuação do arquivo que está armazenado nos *i-nodes* 45 e 46 apresentados nos exemplos anteriores.

Essa ferramenta é muito especial para uma investigação forense, pois é ela que vai exibir o conteúdo que está armazenado na mídia explorada. Esse estudo de caso mostra com clareza o conteúdo dos *i-nodes* 45, 46 e 47, que por consequência, utilizando outra ferramenta, nos mostra que os *i-nodes* estão livres. Esse tipo de sistemas de arquivos consiste em que as informações armazenadas permaneçam ali gravadas, mas o que muda é a informação dos estados dos *i-nodes*, pois quando outra informação é armazenada o sistema de arquivos verifica na tabela quais *i-nodes* estão livres, e depois armazena o mesmo. Com isso a nova informação sobrescreve a outra.

Pode-se concluir com esse estudo de caso que essas informações dos *i-nodes* envolvidos nesse estudo caso (45, 46 e 47 e os demais) vão ser apagados conforme novas informações forem sendo armazenadas. No próximo estudo de caso são apresentadas duas ferramentas que exploram basicamente as características dos arquivos em relação ao tempo de criação, modificação e acesso. Explorando alguns atributos dos arquivos.

```
# unrm -e -f ext2fs ext2disk.img 47
```

ebido para ser simples de configurar e
usar em redes pequenas. Porém, essa aparente simplicidade desaparece
em qualquer instalação que contém duas ou mais redes Ethernet. É o
que
eu chamo de "curva de aprendizado exponencial", característica que o
NetBIOS compartilha com o Windows, Visual Basic e assim por diante.
Outros protocolos (TCP/IP) e softwares (Linux) obrigam seus usuários a
ter um conhecimento relativamente profundo por mais simples que seja
a
tarefa; porém não reservam "surpresas" para instalações de
complexidade crescente. Portanto, desenham uma curva de aprendizado
logarítmica, de que tanto gostam os Linuxers.

</p><p align="justify">
Não obstante, o NetBios - assim como o DHCP, para quem o conhece -
tem
o seu valor, e, uma vez dominados seus conceitos, pode ser útil e
muito adequado até mesmo em redes sem máquinas Windows.

</p><p align="justify">
Este documento não subs

Figura 9 – Usando o aplicativo Unrm com argumentos `-e -f ext2fs` com I-node 47.

7.3 ESTUDO DE CASO Nº 03 – UTILIZANDO O GRAVE-ROBBER COM MACTIME PARA APRESENTAR INFORMAÇÕES DE DATAS DE MODIFICAÇÃO, ACESSO E CRIAÇÃO DOS ARQUIVOS.

Neste estudo de caso são apresentadas duas ferramentas bastante importantes para determinar as especificações de horários e datas da criação dos arquivos. Como dito anteriormente, alguns dos atributos que são armazenados junto a todos arquivos gravados no computador, são as datas e horários de criação, acesso e modificação. O *grave-robber*, junto com o *mactime*, irão coletar e armazenar essas informações em determinados arquivos conforme seu uso, com isso fazendo uma cronologia destas características (horários e datas) dos arquivos envolvidos. Para utilizar essas ferramentas precisa ter acessos a mesmas. Para se ter acesso aos arquivos da imagem do disquete, foi criado um diretório chamado de *mntforensic* que é o ponto de montagem da imagem. Para montar uma imagem de disco, utiliza-se o aplicativo *mount* que é instalado por padrão no Debian GNU/Linux. O aplicativo foi utilizado desta maneira:

```
# mount -o ro ext2disk.img mntforensic
```

Figura 10 – Montando a imagem do disquete no diretório mntforensic.

Desta maneira monta-se à imagem no diretório que foi utilizado para as ferramentas descritas neste estudo de caso. O argumento *-o ro* significa que a imagem do disquete vai ser utilizada como somente leitura (*read-only*).

Na figura 11, com a imagem já montada, foi utilizado o *grave-robber*, com alguns argumentos. O argumento *-m* indica que o resultado do *grave-robber* vai ser utilizado como fonte de informação para a ferramenta *mactime*, significa que a mesma funciona a partir dos resultados do *grave-robber*. Já o argumento *-b coleta* indica que o resultado será armazenado num arquivo chamando *coleta*, outro argumento que deve ser passado é o diretório onde estão os arquivos, no diretório *mntforensic*. Pode-se observar que foi gerada uma tabela, indicando entradas dos arquivos que estão armazenados naquela imagem. Estão destacados em negrito alguns pontos que o docente pode apresentar para os alunos. Como por exemplo, as informações do local e nome dos arquivos, o *i-node* alocado para o armazenamento do mesmo, as permissões de acessos, que é característica do sistema de arquivos *ext2*.

No sistema de arquivos *ext2* as permissões dos arquivos são implementadas com bits de acessos. O sistema de arquivos oferece três bits para o proprietário do arquivo, mais três

```
# grave-robber -m -b coleta mntforensic

class|host|start_time
body|n3w5|1125049615
md5|file|st_dev|st_ino|st_mode|st_ls|st_nlink|st_uid|st_gid|st_rdev|st_size|st_a
time|st_mtime|st_ctime|st_blksize|st_blocks
000|mntforensic/lost+found|1793|11|16832|drwx-----
|2|0|0|0|1024|1124275549|1124275396|1124275396|4096|2
000|mntforensic/ConfiguracaoAna.pdf|1793|12|33188|-rw-r--r--
|1|0|0|0|11718|1124275439|1124275440|1124275440|4096|24
000|mntforensic/samba_intro.htm|1793|13|33188|-rw-r--r--
|1|0|0|0|36861|1124275440|1124275440|1124275440|4096|74
000|mntforensic/ch-apend.htm|1793|14|33261|-rwxr-xr-
x|1|0|0|0|15150|1124275485|1124275485|1124275485|4096|32
000|mntforensic/ch-bas.htm|1793|15|33261|-rwxr-xr-
x|1|0|0|0|11080|1124275485|1124275485|1124275485|4096|22
000|mntforensic/ch-cfgrede.htm|1793|16|33261|-rwxr-xr-
x|1|0|0|0|30475|1124275485|1124275485|1124275485|4096|62
000|mntforensic/ch-fw-iptables.htm|1793|21|33261|-rwxr-xr-
x|1|0|0|0|143512|1124275485|1124275485|1124275485|4096|284
000|mntforensic/ch-hardw.htm|1793|22|33261|-rwxr-xr-
x|1|0|0|0|94554|1124275485|1124275485|1124275485|4096|188
000|mntforensic/ch-impr.htm|1793|23|33261|-rwxr-xr-
x|1|0|0|0|22263|1124275485|1124275485|1124275485|4096|46
000|mntforensic/ch-intro.htm|1793|24|33261|-rwxr-xr-
x|1|0|0|0|21399|1124275485|1124275485|1124275485|4096|44
000|mntforensic/ch-log.htm|1793|25|33261|-rwxr-xr-
x|1|0|0|0|28368|1124275485|1124275485|1124275485|4096|58
000|mntforensic/ch-pers.htm|1793|26|33261|-rwxr-xr-
x|1|0|0|0|14035|1124275485|1124275485|1124275485|4096|30
000|mntforensic/ch-rede.htm|1793|27|33261|-rwxr-xr-
x|1|0|0|0|60419|1124275485|1124275485|1124275485|4096|122
000|mntforensic/ch-s-apache.htm|1793|28|33261|-rwxr-xr-
x|1|0|0|0|174183|1124275485|1124275485|1124275485|4096|344
000|mntforensic/ch-s-cvs.htm|1793|29|33261|-rwxr-xr-
x|1|0|0|0|60625|1124275485|1124275485|1124275485|4096|122
000|mntforensic/ch-s-ident.htm|1793|30|33261|-rwxr-xr-
x|1|0|0|0|16757|1124275485|1124275485|1124275485|4096|36
000|mntforensic/ch-s-pop3.htm|1793|31|33261|-rwxr-xr-
x|1|0|0|0|16353|1124275485|1124275485|1124275485|4096|34
000|mntforensic/ch-s-samba.htm|1793|32|33261|-rwxr-xr-
x|1|0|0|0|215721|1124275485|1124275485|1124275485|4096|424
000|mntforensic/ch-s-ssh.htm|1793|33|33261|-rwxr-xr-
x|1|0|0|0|45583|1124275485|1124275485|1124275485|4096|92
000|mntforensic/ch-s-telnet.htm|1793|34|33261|-rwxr-xr-
x|1|0|0|0|10888|1124275485|1124275485|1124275485|4096|22
000|mntforensic/index.htm|1793|35|33261|-rwxr-xr-
x|1|0|0|0|67293|1124275485|1124275485|1124275485|4096|134
```

Figura 11 – Usando grave-robber para uma coleta de informações.

bits para o grupo onde o usuário faz parte e os restantes três bits são atribuídos para os outros usuários. O padrão desses bits é representado pelas letras *rwx*, onde são os acessos de *read* (leitura), *write* (gravação) e *execute* (execução) respectivamente.

Para cada usuário, é atribuído um identificador, o *uid*, e outro identificador para o grupo onde o mesmo faz parte, o *gid*. Com essas características, o sistema de arquivos pode implementar os nove bits para cada arquivo. Outro bit é bastante importante, que identifica que tipo de arquivo é, como por exemplo, na figura 11 o arquivo *lost+found* que está alocado no *i-node* 11 é do tipo diretório, pois o campo *st_ls* da tabela do *grave-robber* representa as permissões do diretório, e o primeiro bit é *d* identificando que é um diretório. Para os arquivos de comuns (textos e imagens) é representado pelo sinal de “-” (menos). Com todas essas características o docente pode explorar pontos importantes dos

assuntos envolvidos no sistema de arquivos.

Na figura 12, mostra uma visão clara da cronologia do tempo das criações, modificações e acessos dos arquivos pertencentes à imagem do disquete. O *mactime* foi utilizado com alguns argumentos para o objetivo deste estudo de caso. Como dito anteriormente, o *mactime* utiliza o arquivo que é gerado a parti do *grave-robbber* como base para seu resultado. Então na figura 11 o arquivo *coleta* que o *grave-robbber* gerou foi aproveitado como arquivo base para a produção do *mactime*, nesta figura 12.

```
# mactime -h -b coleta 08/17/2005
```

Aug 17 05 10:43:16	1024	m.c	drwx-----	root	root	mntforensic/lost+found
Aug 17 05 10:43:59	11718	.a.	-rw-r--r--	root	root	mntforensic/ConfiguracaoAna.pdf
Aug 17 05 10:44:00	36861	mac	-rw-r--r--	root	root	
mntforensic/samba						intro.htm
Aug 17 05 10:44:45	16353	mac	-rwxr-xr-x	root	root	mntforensic/ch-s-pop3.htm
	22263	mac	-rwxr-xr-x	root	root	mntforensic/ch-impr.htm
	94554	mac	-rwxr-xr-x	root	root	mntforensic/ch-hardw.htm
	174183	mac	-rwxr-xr-x	root	root	mntforensic/ch-s-apache.htm
	67293	mac	-rwxr-xr-x	root	root	mntforensic/index.htm
	28368	mac	-rwxr-xr-x	root	root	mntforensic/ch-log.htm
	10888	mac	-rwxr-xr-x	root	root	mntforensic/ch-s-telnet.htm
	215721	mac	-rwxr-xr-x	root	root	mntforensic/ch-s-samba.htm
	14035	mac	-rwxr-xr-x	root	root	mntforensic/ch-pers.htm
	15150	mac	-rwxr-xr-x	root	root	mntforensic/ch-apend.htm
	60419	mac	-rwxr-xr-x	root	root	mntforensic/ch-rede.htm
	16757	mac	-rwxr-xr-x	root	root	mntforensic/ch-s-ident.htm
	143512	mac	-rwxr-xr-x	root	root	mntforensic/ch-fw-iptables.htm
	21399	mac	-rwxr-xr-x	root	root	mntforensic/ch-intro.htm
	45583	mac	-rwxr-xr-x	root	root	mntforensic/ch-s-ssh.htm
	60625	mac	-rwxr-xr-x	root	root	mntforensic/ch-s-cvs.htm
	11080	mac	-rwxr-xr-x	root	root	mntforensic/ch-bas.htm
	30475	mac	-rwxr-xr-x	root	root	mntforensic/ch-cfgrede.htm
Aug 17 05 10:45:49	1024	.a.	drwx-----	root	root	mntforensic/lost+found

Figura 12 – Usando o aplicativo Mactime para apresentar os atributos das datas de criação, modificação e acesso dos arquivos.

O argumento passado para a ferramenta, na figura 12, consiste em que o *-h* significa para apresentar o resultado em um arquivo de *html* simples, ao invés de texto em ASCII puro. Outro argumento é o *-b coleta* que significa para o *mactime* utilizar o arquivo *coleta* como fonte de informações sobre a imagem do disquete, e o último argumento é a data 08/17/2005 que solicita para a ferramenta apresentar todos os arquivos criados, acessados e modificados de acordo com esta data. O *mactime* utiliza o padrão de data *MM/DD/YYYY* onde o *MM* representa o mês, *DD* o dia e o *YYYY* o ano.

O docente pode utilizar este estudo de caso, para mostrar especificamente os atributos das datas de criação, modificação e acesso dos arquivos apresentados, fazendo sempre uma abordagem prática o assunto estudado. A terceira coluna indica os estados dos arquivos.

7.4 PROPOSTA DE AULA DE SISTEMAS DE ARQUIVOS UTILIZANDO FERRAMENTAS FORENSE.

A seguir será apresentada uma aula com o uso das ferramentas forense para auxiliar no processo de aprendizado do aluno. Será apresentada uma breve introdução, descrevendo suas principais características em relação a propriedades e implementações do sistema de arquivos utilizado na pesquisa (*ext2*), descrevendo seus métodos e em que situações as ferramentas forense tem relação com os assuntos apresentados. Os estudos de casos podem ajudar na compreensão do uso das ferramentas pelo ao aluno, e em relação aos tópicos do assunto.

Tomando como base teórica o assunto sistema de arquivos, tomou-se como referência um livro do professor Andrew S. Tanenbaum (2000). O autor explica os princípios e características existentes em um SO *Minix*, utilizado por um de seus alunos para desenvolver um dos SO mais robustos e seguros da atualidade, o *Linux*.

7.5 UTILIZANDO AS FERRAMENTAS ILS E UNRM PARA MELHORAR A COMPREENSÃO DO ALUNO NO ASSUNTO IMPLEMENTAÇÕES DE SISTEMAS DE ARQUIVOS.

Segundo Tanenbaum (2000), tratando-se da forma que é implementado o sistema de arquivos, o usuário comum tem a preocupação de como é organizado, pelo SO, a gravação e a recuperação dos seus arquivos. Ao nível de usuário estudante de informática, existe essa preocupação também, pois o mesmo não deixa de ser usuário, e existe uma outra preocupação (curiosidade) em relação de como o SO gerencia e controla a gravação e a recuperação de arquivos. No Unix, a forma de implementação é através do método de associar com cada arquivo um endereço de tabela, chamada de Nós-i (tabela de *i-node*), onde o mesmo armazena seus atributos e endereço do disco onde está localizado (número do *i-nodes*). No estudo de caso número 01, com o uso da ferramenta *ils*, na figura 04, observa-se que tais atributos são armazenados de forma organizada pelo sistema de arquivos utilizado na pesquisa. O docente explora alguns campos existentes na tabela, como por exemplo, o campo do número da entrada do arquivo na tabela (ST_INO), pois é com esse número que o sistema de arquivo localiza os arquivos gravados e com isso pode recuperar para o usuário, o campo onde identifica o proprietário do arquivo (ST_UID), outro campo pode ser observado e do tamanho do arquivo (ST_SIZE), dentre outros campos. O docente pode explorar a cognição do aluno e deixar o próprio observar a tabela e tentar relacionar com a teoria. Com isso o aluno terá oportunidade de apreender de forma clara como é armazenado o arquivo. No nível de estudante de informática é de extrema importância científica estudar com mais detalhes a

forma de implementar e organizar o funcionamento do SO pesquisado.

A forma que é implementada pelo SO, utilizada na pesquisa, para a recuperação de arquivos é através dos números da entrada na tabela de *i-nodes*. Utilizando o estudo de caso número 02, com o uso da ferramenta *unrm*, pode observar que nas figuras 07, 08 e 09 apresenta um arquivo do tipo *html* (*HiperText Markup Language*) que está na tabela de *i-nodes*, com o campo de (ST_ALLOC) “f”, significando que essa entrada na tabela está vazia. A ferramenta utiliza o número da entrada na tabela para processar a solicitação, então o que o sistema de arquivos precisa, de grande importância, é do número do *i-nodes* daquele arquivo solicitado para fazer o processo de recuperação e poder carregá-lo para a memória. O docente nesta hora pode fazer uma pergunta, do tipo, a entrada na tabela está indicando que está livre, porque apresenta o conteúdo do arquivo? O docente explica que quando um arquivo é deletado, o que é processado pelo sistema de arquivos é um comando para retirar a entrada do arquivo na tabela de *i-nodes*.

Conforme Tanenbaum (2000), a implementação na tabela de *i-nodes* segue da seguinte forma: quando o arquivo é pequeno o sistema de arquivos armazena em uma única entrada na tabela, quando o arquivo é grande ele armazena em diversas entradas até que o arquivo seja todo armazenado na tabela, nesta implementação a tabela utiliza um campo (ST_NLINK, pode observar na figura 04) para especificar o lugar onde está localizada a próxima entrada restante do arquivo. Tanenbaum utiliza os nomes de bloco indireto simples para se referenciar uma única entrada na tabela de *i-nodes*, quando o arquivo utiliza duas entradas e é chamado de bloco indireto duplo, e assim por diante.

7.6 UTILIZANDO A FERRAMENTA GRAVER-ROBBER E MACTIME PARA MELHORAR A COMPREENSÃO DO ALUNO NO ASSUNTO ATRIBUTOS DE ARQUIVOS.

Retomando Tanenbaum (2000), a necessidade de se criar um componente para armazenamento dos conteúdos que eram processados pelo computador, deu início ao sistema de arquivos, pois em computadores antigos não tinha um componente de armazenagem contido no mesmo. O usuário tinha diversos problemas neste sentido, por exemplo, caso ele quisesse processar um aplicativo e armazenar o resultado, teria que anotar em outro lugar, pois quando o processo terminava, seus valores, atributos da aplicação inclusive os resultados iriam desaparecer. Então houve a necessidade do armazenamento, pois o usuário, posteriormente queria acessar essa mesma informação. A solução era armazenar em mídias externas ou em discos em unidades chamada arquivos.

No sistema de arquivos utilizado na pesquisa, existem diversas características, como por exemplo, em relação ao nome dos arquivos que há uma distinção de nome em letras maiúsculas e minúsculas. Outro ponto importante é que o SO só sabe o que são bytes, e armazena em uma sequência organizada, como vimos anteriormente. Sabemos que existem diversos tipos de arquivos, que pode ser arquivos comuns, que armazena informações de usuário, diretórios que é uma maneira que o SO tem para organizar hierarquicamente seus arquivos, arquivos especiais de caracteres e arquivos de especiais de bloco, utilizados para modelar disco.

Cada arquivo contém alguns atributos como, por exemplo, data/hora de criação, o identificador do seu proprietário, o tamanho em bytes, as permissões de leitura/gravação, data/hora de modificação, data/hora de acesso, dentre outros possíveis atributos.

Com a ferramenta utilizada no estudo de caso número 03, o *grave-robber* e o *mactime*, o docente apresenta de maneira clara (veja figura 11), alguns atributos importantes, como por exemplo, o nome do arquivo, a identificação do número do *i-nodes* que está alocado, se um arquivo é comum ou do tipo diretório, a identificação do usuário pelo seu *uid*.

Na figura 11 novamente, apresenta um campo importante que é das permissões que o arquivo têm, em relação à leitura/gravação que é característica do sistema de arquivo utilizado na pesquisa, que é os nove bits que define essas permissões.

Na figura 12 a ferramenta *mactime* mostra com clareza os atributos dos arquivos em relação à data/hora de criação, de modificação e acesso. O docente explica a utilização desta ferramenta para apresentar que quando um arquivo é armazenado, o sistema utiliza a data/hora do SO para gravar no arquivo o atributo da data/hora de criação, e que na figura 12, está mostrando todos os arquivos criados, modificados ou acessados no dia 08/17/2005 (utilizando o padrão mês/dia/ano).

Observando o arquivo “*ConfiguracaoAna.pdf*” da figura 12, pode dizer que naquele dia este arquivo foi acessado, pois a ferramenta *mactime* atribui no seu resultado a letra “*.a.*” significa *access* (acessado), e observando outro arquivo, por exemplo, o “*index.htm*” a ferramenta atribui no seu resultado as letras “*m.a.c.*”, significando *modification, access e ciation* (modificado, acessado e criado). O docente mostra que a ferramenta coleta esses atributos armazenados junto aos arquivos que iram identificá-los dos demais.

8 ANÁLISE DE RESULTADOS

Com a realização deste projeto, foi possível partir para o experimento prático, de acordo com a metodologia apresentada no projeto de monografia, de modo exploratório. Para atingir os objetivos propostos, a técnica exploratória foi a realização de uma aula da disciplina de sistemas operacionais com o uso das ferramentas da forense computacional, descritas no desenvolvimento, auxiliando na explicação do docente em torno do assunto sistemas de arquivos. O experimento teve a duração de 45 minutos, descrevendo determinadas características do sistema de arquivos, apresentando em paralelo propostas práticas do uso das ferramentas envolvidas.

As ferramentas *ils* e *unrm* foram utilizadas para auxiliar o docente a apresentar certas características da implementação (gravação e recuperação de arquivos) da tabelas de inodes do sistemas de arquivos *ext2* (padrão *unix*). As ferramentas *grave-robber* e *mactime* para auxiliar o docente a apresentar características existentes em arquivos.

O experimento ocorreu no laboratório de redes da FARN (Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte) com máquinas equipadas com Sistema Operacional Debian GNU/Linux (versão estável), com todas as ferramentas descritas no desenvolvimento da monografia. Cada aluno no seu micro aplicando o método descrito no desenvolvimento da monografia, em paralelo, o docente explicando conceitos da disciplina de sistemas operacionais, com isso relacionando teoria com a prática. Com esse método, é possível afirmar que os alunos ficaram bastante interessados com o uso das ferramentas, por que mostrando as estruturas e características do funcionamento do sistema operacional, que é fundamental para o aprendizado dos alunos. Cada aluno possui um perfil de estudante que já estudou a disciplina de sistemas operacionais.

Ao término do experimento foi realizado um exame de aceitação da metodologia aplicada ao ensino de sistemas operacionais, contendo três perguntas objetivas de duas opções SIM ou NÃO. Depurando as respostas pode-se observar que todos os alunos envolvidos no experimento prático, recomendam o uso do método como uma metodologia para o ensino da disciplina de sistemas operacionais.

9 CONCLUSÃO

Nesta Monografia foi feita a descrição das principais características da Forense Computacional e de alguns módulos que compõem um sistema operacional envolvido na realização e conclusão deste trabalho. Com a pesquisa realizada foi possível identificar certos pontos positivos e negativos da metodologia tradicional no ensino da disciplina de sistemas operacionais. Utilizando as ferramentas disponíveis no próprio sistema operacional, ao nível de usuário comum, o próprio oculta certas características que são fundamentais para uma melhor compreensão do funcionamento do mesmo. Ao nível de usuário estudante, os sistemas operacionais deixam de apresentar certas estruturas e características do seu funcionamento, que são de extrema importância para o ensino e aprendizado do aluno sobre a disciplina de sistemas operacionais. Uma possível dificuldade com uso deste método é a aprendizagem do uso do sistema operacional Debian GNU/Linux e das ferramentas, por parte do docente e dos alunos.

De acordo com o experimento prático é possível concluir que com o uso das aplicações da forense computacional como ferramenta de apoio ao ensino da disciplina sistemas operacionais, pode-se conseguir um maior nível de aceitação por parte dos alunos envolvidos e conseqüentemente melhorando o ensino e aprendizado do mesmo, sempre trabalhando uma metodologia com o uso da prática da docência.

10 REFERÊNCIAS BIBLIOGRÁFICAS

REIS, Marcelo Abdalla. **Forense Computacional: Procedimentos e Padrões**. Instituto de Computação - UNICAMP. Campinas, Universidade Estadual de Campinas, 2004.

ATÍLO, César Eduardo. **Padrão "ACME!" Para Análise Forense de Intrusões em Sistemas Computacionais**. UNESP, 2003.

FREITAS, Andrey Rodrigues. **Perícia Forense Aplicada à Informática**. IBPI, 2003.

FREIRE, Paulo. **Educação e Mudança**. Rio de Janeiro: Paz e Terra, 1985.

Crimes de Informática **Pedagogia da Autonomia**. Rio de Janeiro: Paz e Terra, 1997.

GEUS, Paulo Lício de. **Modagem de um Sistema Automatizado de Análise Forense: Arquitetura Extensível e Protótipo Inicial**. Instituto de Computação - UNICAMP.

MIEL, A. **Criatividade no Ensino**. São Paulo: IBRASA, 1972.

MINAYO, Maria Cecília de Souza. **Pesquisa Social: Teoria, Método e Criatividade**. Petrópolis: Editora Vozes, 1997

SEYMOUR, Paper. **A Máquina das Crianças: Repensando a Escola na Era da Informática**. Tradução Sandra Costa. Porto Alegre-RS: Artmed, 2002.

TANENBAUM, Andrew S. **Sistema Operacionais: projeto e implementação**. 2 ed. Porto Alegre: Bookman, 2000.

11 BIBLIOGRAFIA CONSULTADA

COSTA, Cristiano Reinaldo Itapoan da. **Sistemas para Grupos de Respostas a Incidentes de Segurança em Computadores**. Universidade Luterana do Brasil, 2004.

DHANJANI, Nitesh. **Segurança no Linux e Unix**. Rio de Janeiro: Campus, 2003.

GUIMARÃES, Célio Cardoso. **Forense Computacional: Aspectos Legais e Padronização**. Instituto de Computação - UNICAMP.

JESSEN, Klaus Steding. **Evidências em Sistemas Unix**. I Seminário Nacional de Perícia em Crimes de Informática, 2002.

OLIVEIRA, Flávio. **Metodologias de Análise Forense para Ambientes Baseados em NTFS**. III Simpósio de Segurança em Informática, 2001.

REIS, Marcelo Abdalla dos. **Análise Forense e Intrusões em Sistemas Computacionais: Técnicas, Procedimentos e Ferramentas**. Instituto de Computação Estadual de Campinas – SP. 2002.

SERAFIM, Vinicius da Silveira. **Um Verificador Seguro de Integridade de Arquivos**. III Simpósio de Segurança em Informática, 2001.

TANENBAUM, Andrew S. **Sistema Operacional: projeto e implementação**. 2 ed. Porto Alegre: Bookman, 2000.

LICENÇA PÚBLICA GERAL GNU

Versão 2, Junho de 1991.

This is an unofficial translation of the GNU General Public License into Brazilian Portuguese.

It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL -- only the original English text of the GNU GPL does that. However, we hope that this translation will help Brazilian Portuguese speakers understand the GNU GPL better.

Esta é uma tradução não-oficial da Licença Pública Geral GNU ("GPL GNU") para o português do Brasil. Ela não foi publicada pela Free Software Foundation, e legalmente não afirma os termos de distribuição de software que utiliza a GPL GNU -- apenas o texto original da GPL GNU, em inglês, faz isso. Contudo, esperamos que esta tradução ajude aos que utilizam o português do Brasil a entender melhor a GPL GNU.

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave,

Cambridge, MA 02139, USA

ANEXO A

A qualquer pessoa é permitido copiar e distribuir cópias desse documento de licença, desde que sem qualquer alteração.

INTRODUÇÃO

As licenças de muitos software são desenvolvidas para restringir sua liberdade de compartilhá-lo e mudá-lo. Contrária a isso, a Licença Pública Geral GNU pretende garantir sua liberdade de compartilhar e alterar software livres -- garantindo que o software será livre e gratuito para os seus usuários. Esta Licença Pública Geral aplica-se à maioria dos software da Free Software Foundation e a qualquer outro programa cujo autor decida aplicá-la. (Alguns outros software da FSF são cobertos pela Licença Pública Geral de Bibliotecas, no entanto.) Você pode aplicá-la também aos seus programas.

Quando nos referimos a software livre, estamos nos referindo a liberdade e não a preço. Nossa Licença Pública Geral foi desenvolvida para garantir que você tenha a liberdade de distribuir cópias de software livre (e cobrar por isso, se quiser); que você receba o código

LICENÇA PÚBLICA GERAL GNU

Versão 2, junho de 1991

This is an unofficial translation of the GNU General Public License into Brazilian Portuguese.

It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL -- only the original English text of the GNU GPL does that. However, we hope that this translation will help Brazilian Portuguese speakers understand the GNU GPL better.

Esta é uma tradução não-oficial da Licença Pública Geral GNU ("GPL GNU") para o português do Brasil. Ela não foi publicada pela Free Software Foundation, e legalmente não afirma os termos de distribuição de software que utiliza a GPL GNU -- apenas o texto original da GPL GNU, em inglês, faz isso. Contudo, esperamos que esta tradução ajude aos que utilizam o português do Brasil a entender melhor a GPL GNU.

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave,

Cambridge, MA 02139, USA

A qualquer pessoa é permitido copiar e distribuir cópias desse documento de licença, desde que sem qualquer alteração.

INTRODUÇÃO

As licenças de muitos software são desenvolvidas para restringir sua liberdade de compartilhá-lo e mudá-lo. Contrária a isso, a Licença Pública Geral GNU pretende garantir sua liberdade de compartilhar e alterar software livres -- garantindo que o software será livre e gratuito para os seus usuários. Esta Licença Pública Geral aplica-se à maioria dos software da Free Software Foundation e a qualquer outro programa cujo autor decida aplicá-la. (Alguns outros software da FSF são cobertos pela Licença Pública Geral de Bibliotecas, no entanto.) Você pode aplicá-la também aos seus programas.

Quando nos referimos a software livre, estamos nos referindo a liberdade e não a preço. Nossa Licença Pública Geral foi desenvolvida para garantir que você tenha a liberdade de distribuir cópias de software livre (e cobrar por isso, se quiser); que você receba o código-

fonte ou tenha acesso a ele, se quiser; que você possa mudar o software ou utilizar partes dele em novos programas livres e gratuitos; e que você saiba que pode fazer tudo isso.

Para proteger seus direitos, precisamos fazer restrições que impeçam a qualquer um negar estes direitos ou solicitar que você deles abdique. Estas restrições traduzem-se em certas responsabilidades para você, se você for distribuir cópias do software ou modificá-lo.

Por exemplo, se você distribuir cópias de um programa, gratuitamente ou por alguma quantia, você tem que fornecer aos recebedores todos os direitos que você possui. Você tem que garantir que eles também recebam ou possam obter o código-fonte. E você tem que mostrar-lhes estes termos para que eles possam conhecer seus direitos.

Nós protegemos seus direitos em dois passos: (1) com copyright do software e (2) com a oferta desta licença, que lhe dá permissão legal para copiar, distribuir e/ou modificar o software.

Além disso, tanto para a proteção do autor quanto a nossa, gostaríamos de certificarmos que todos entendam que não há qualquer garantia nestes software livres. Se o software é modificado por alguém mais e passado adiante, queremos que seus recebedores saibam que o que eles obtiveram não é original, de forma que qualquer problema introduzido por terceiros não interfira na reputação do autor original.

Finalmente, qualquer programa é ameaçado constantemente por patentes de software. Queremos evitar o perigo de que distribuidores de software livre obtenham patentes individuais, o que tem o efeito de tornar o programa proprietário. Para prevenir isso, deixamos claro que qualquer patente tem que ser licenciada para uso livre e gratuito por qualquer pessoa, ou então que nem necessite ser licenciada.

Os termos e condições precisas para cópia, distribuição e modificação se encontram abaixo:

LICENÇA PÚBLICA GERAL GNU

TERMOS E CONDIÇÕES PARA CÓPIA, DISTRIBUIÇÃO E MODIFICAÇÃO.

0. Esta licença se aplica a qualquer programa ou outro trabalho que contenha um aviso colocado pelo detentor dos direitos autorais informando que aquele pode ser distribuído sob as

condições desta Licença Pública Geral. O "Programa" abaixo refere-se a qualquer programa ou trabalho, e "trabalho baseado no Programa" significa tanto o Programa em si como quaisquer trabalhos derivados, de acordo com a lei de direitos autorais: isto quer dizer um trabalho que contenha o Programa ou parte dele, tanto originalmente ou com modificações, e/ou tradução para outros idiomas. (Doravante o processo de tradução está incluído sem limites no termo "modificação".) Cada licenciado é mencionado como "você".

Atividades outras que a cópia, a distribuição e modificação não estão cobertas por esta Licença; elas estão fora de seu escopo. O ato de executar o Programa não é restringido e o resultado do Programa é coberto apenas se seu conteúdo contenha trabalhos baseados no Programa (independentemente de terem sido gerados pela execução do Programa). Se isso é verdadeiro depende do que o programa faz.

1. Você pode copiar e distribuir cópias fiéis do código-fonte do Programa da mesma forma que você o recebeu, usando qualquer meio, desde que você conspícua e apropriadamente publique em cada cópia um aviso de direitos autorais e uma declaração de inexistência de garantias; mantenha intactas todos os avisos que se referem a esta Licença e à ausência total de garantias; e forneça a outros recebedores do Programa uma cópia desta Licença, junto com o Programa. Você pode cobrar pelo ato físico de transferir uma cópia e pode, opcionalmente, oferecer garantia em troca de pagamento.

2. Você pode modificar sua cópia ou cópias do Programa, ou qualquer parte dele, assim gerando um trabalho baseado no Programa, e copiar e distribuir essas modificações ou trabalhos sob os termos da seção 1 acima, desde que você também se enquadre em todas estas condições:

a) Você tem que fazer com que os arquivos modificados levem avisos proeminentes afirmando que você alterou os arquivos, incluindo a data de qualquer alteração.

b) Você tem que fazer com que quaisquer trabalhos que você distribua ou publique, e que integralmente ou em partes contenham ou sejam derivados do Programa ou de suas partes, sejam licenciados, integralmente e sem custo algum para quaisquer terceiros, sob os termos desta Licença.

c) Se qualquer programa modificado normalmente lê comandos interativamente quando executados, você tem que fazer com que, quando iniciado tal uso interativo da forma mais

simples, seja impresso ou mostrado um anúncio de que não há qualquer garantia (ou então que você fornece a garantia) e que os usuários podem redistribuir o programa sob estas condições, ainda informando os usuários como consultar uma cópia desta Licença. (Exceção: se o Programa em si é interativo mais normalmente não imprime estes tipos de anúncios, seu trabalho baseado no Programa não precisa imprimir um anúncio.)

Estas exigências aplicam-se ao trabalho modificado como um todo. Se seções identificáveis de tal trabalho não são derivadas do Programa, e podem ser razoavelmente consideradas trabalhos independentes e separados por si só, então esta Licença, e seus termos, não se aplicam a estas seções quando você distribui-las como trabalhos em separado. Mas quando você distribuir as mesmas seções como parte de um todo que é trabalho baseado no Programa, a distribuição como um todo tem que se enquadrar nos termos desta Licença, cujas permissões para outros licenciados se estendem ao todo, portanto também para cada e toda parte independente de quem a escreveu.

Desta forma, esta seção não tem a intenção de reclamar direitos ou contestar seus direitos sobre o trabalho escrito completamente por você; ao invés disso, a intenção é a de exercitar o direito de controlar a distribuição de trabalhos, derivados ou coletivos, baseados no Programa.

Adicionalmente, a mera adição ao Programa de outro trabalho não baseado no Programa (ou de trabalho baseado no Programa) em um volume de armazenamento ou meio de distribuição não faz o outro trabalho parte do escopo desta Licença.

3. Você pode copiar e distribuir o Programa (ou trabalho baseado nele, conforme descrito na Seção 2) em código-objeto ou em forma executável sob os termos das Seções 1 e 2 acima, desde que você faça um dos seguintes:

- a) O acompanhe com o código-fonte completo e em forma acessível por máquinas, que tem que ser distribuído sob os termos das Seções 1 e 2 acima e em meio normalmente utilizado para o intercâmbio de software; ou,
- b) O acompanhe com uma oferta escrita, válida por pelo menos três anos, de fornecer a qualquer um, com um custo não superior ao custo de distribuição física do material, uma cópia do código-fonte completo e em forma acessível por máquinas, que tem que ser

distribuído sob os termos das Seções 1 e 2 acima e em meio normalmente utilizado para o intercâmbio de software; ou,

c) O acompanhe com a informação que você recebeu em relação à oferta de distribuição do código-fonte correspondente. (Esta alternativa é permitida somente em distribuição não comerciais, e apenas se você recebeu o programa em forma de código-objeto ou executável, com oferta de acordo com a Subseção b acima.)

O código-fonte de um trabalho corresponde à forma de trabalho preferida para se fazer modificações. Para um trabalho em forma executável, o código-fonte completo significa todo o código-fonte de todos os módulos que ele contém, mais quaisquer arquivos de definição de "interface", mais os "scripts" utilizados para se controlar a compilação e a instalação do executável. Contudo, como exceção especial, o código-fonte distribuído não precisa incluir qualquer componente normalmente distribuído (tanto em forma original quanto binária) com os maiores componentes (o compilador, o "kernel" etc.) do sistema operacional sob o qual o executável funciona, a menos que o componente em si acompanhe o executável.

Se a distribuição do executável ou código-objeto é feita através da oferta de acesso a cópias de algum lugar, então ofertar o acesso equivalente a cópia, do mesmo lugar, do código-fonte equivale à distribuição do código-fonte, mesmo que terceiros não sejam compelidos a copiar o código-fonte com o código-objeto.

4. Você não pode copiar, modificar, sub-licenciar ou distribuir o Programa, exceto de acordo com as condições expressas nesta Licença. Qualquer outra tentativa de cópia, modificação, sub-licenciamento ou distribuição do Programa não é válida, e cancelará automaticamente os direitos que lhe foram fornecidos por esta Licença. No entanto, terceiros que de você receberam cópias ou direitos, fornecidos sob os termos desta Licença, não terão suas licenças terminadas, desde que permaneçam em total concordância com ela.

5. Você não é obrigado a aceitar esta Licença já que não a assinou. No entanto, nada mais o dará permissão para modificar ou distribuir o Programa ou trabalhos derivados deste. Estas ações são proibidas por lei, caso você não aceite esta Licença. Desta forma, ao modificar ou distribuir o Programa (ou qualquer trabalho derivado do Programa), você estará indicando sua total aceitação desta Licença para fazê-los, e todos os seus termos e condições para copiar, distribuir ou modificar o Programa, ou trabalhos baseados nele.

6. Cada vez que você redistribuir o Programa (ou qualquer trabalho baseado nele), os recebedores adquirirão automaticamente do licenciador original uma licença para copiar, distribuir ou modificar o Programa, sujeitos a estes termos e condições. Você não poderá impor aos recebedores qualquer outra restrição ao exercício dos direitos então adquiridos. Você não é responsável em garantir a concordância de terceiros a esta Licença.

7. Se, em consequência de decisões judiciais ou alegações de infringimento de patentes ou quaisquer outras razões (não limitadas a assuntos relacionados a patentes), condições forem impostas a você (por ordem judicial, acordos ou outras formas) e que contradigam as condições desta Licença, elas não o livram das condições desta Licença. Se você não puder distribuir de forma a satisfazer simultaneamente suas obrigações para com esta Licença e para com as outras obrigações pertinentes, então como consequência você não poderá distribuir o Programa. Por exemplo, se uma licença de patente não permitirá a redistribuição, livre de "royalties", do Programa, por todos aqueles que receberem cópias direta ou indiretamente de você, então a única forma de você satisfazer a ela e a esta Licença seria a de desistir completamente de distribuir o Programa.

Se qualquer parte desta seção for considerada inválida ou não aplicável em qualquer circunstância particular, o restante da seção se aplica, e a seção como um todo se aplica em outras circunstâncias.

O propósito desta seção não é o de induzi-lo a infringir quaisquer patentes ou reivindicação de direitos de propriedade outros, ou a contestar a validade de quaisquer dessas reivindicações; esta seção tem como único propósito proteger a integridade dos sistemas de distribuição de software livres, o que é implementado pela prática de licenças públicas. Várias pessoas têm contribuído generosamente e em grande escala para os software distribuídos usando este sistema, na certeza de que sua aplicação é feita de forma consistente; fica a critério do autor/doador decidir se ele ou ela está disposto a distribuir software utilizando outro sistema, e um licenciado não pode impor qualquer escolha.

Esta seção destina-se a tornar bastante claro o que se acredita ser consequência do restante desta Licença.

8. Se a distribuição e/ou uso do Programa são restringidos em certos países por patentes ou direitos autorais, o detentor dos direitos autorais original, e que colocou o Programa sob esta Licença, pode incluir uma limitação geográfica de distribuição, excluindo aqueles países de

forma a tornar a distribuição permitida apenas naqueles ou entre aqueles países então não excluídos. Nestes casos, esta Licença incorpora a limitação como se a mesma constasse escrita nesta Licença.

9. A Free Software Foundation pode publicar versões revisadas e/ou novas da Licença Pública Geral de tempos em tempos. Estas novas versões serão similares em espírito à versão atual, mas podem diferir em detalhes que resolvem novos problemas ou situações.

A cada versão é dada um número distinto. Se o Programa especifica um número de versão específico desta Licença que se aplica a ele e a "qualquer nova versão", você tem a opção de aceitar os termos e condições daquela versão ou de qualquer outra versão publicada pela Free Software Foundation. Se o programa não especifica um número de versão desta Licença, você pode escolher qualquer versão já publicada pela Free Software Foundation.

10. Se você pretende incorporar partes do Programa em outros programas livres cujas condições de distribuição são diferentes, escreva ao autor e solicite permissão. Para o software que a Free Software Foundation detém direitos autorais, escreva à Free Software Foundation; às vezes nós permitimos exceções a este caso. Nossa decisão será guiada pelos dois objetivos de preservar a condição de liberdade de todas as derivações do nosso software livre, e de promover o compartilhamento e reutilização de software em aspectos gerais.

AUSÊNCIA DE GARANTIAS

11. UMA VEZ QUE O PROGRAMA É LICENCIADO SEM ÔNUS, NÃO HÁ QUALQUER GARANTIA PARA O PROGRAMA, NA EXTENSÃO PERMITIDA PELAS LEIS APLICÁVEIS. EXCETO QUANDO EXPRESSADO DE FORMA ESCRITA, OS DETENTORES DOS DIREITOS AUTORAIS E/OU TERCEIROS DISPONIBILIZAM O PROGRAMA "NO ESTADO", SEM QUALQUER TIPO DE GARANTIAS, EXPRESSAS OU IMPLÍCITAS, INCLUINDO, MAS NÃO LIMITADO A, AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E AS DE ADEQUAÇÃO A QUALQUER PROPÓSITO. O RISCO TOTAL COM A QUALIDADE E DESEMPENHO DO PROGRAMA É SEU. SE O PROGRAMA SE MOSTRAR DEFEITUOSO, VOCÊ ASSUME OS CUSTOS DE TODAS AS MANUTENÇÕES, REPAROS E CORREÇÕES.

12. EM NENHUMA OCASIÃO, A MENOS QUE EXIGIDO PELAS LEIS APLICÁVEIS OU ACORDO ESCRITO, OS DETENTORES DOS DIREITOS AUTORAIS, OU

QUALQUER OUTRA PARTE QUE POSSA MODIFICAR E/OU REDISTRIBUIR O PROGRAMA CONFORME PERMITIDO ACIMA, SERÃO RESPONSABILIZADOS POR VOCÊ POR DANOS, INCLUINDO QUALQUER DANO EM GERAL, ESPECIAL, ACIDENTAL OU CONSEQÜENTE, RESULTANTES DO USO OU INCAPACIDADE DE USO DO PROGRAMA (INCLUINDO, MAS NÃO LIMITADO A, A PERDA DE DADOS OU DADOS TORNADOS INCORRETOS, OU PERDAS SOFRIDAS POR VOCÊ OU POR OUTRAS PARTES, OU FALHAS DO PROGRAMA AO OPERAR COM QUALQUER OUTRO PROGRAMA), MESMO QUE TAL DETENTOR OU PARTE TENHAM SIDO AVISADOS DA POSSIBILIDADE DE TAIS DANOS.

FIM DOS TERMOS E CONDIÇÕES

ANEXO B

QUESTIONÁRIO

1) Você teve dificuldades no aprendizado da disciplina de Sistemas Operacionais (o mais especificamente sistema de arquivos)?

() SIM

() NÃO

2) A abordagem prática auxilia a compreensão do assunto?

() SIM

() NÃO

3) Você recomendaria a adoção destas ferramentas para o ensino da disciplina de Sistemas Operacionais? Por que?

() SIM

() NÃO

ANEXO B

QUESTIONÁRIO

1) Você teve dificuldades no aprendizado da disciplina de Sistemas Operacionais (e mais especificamente sistema de arquivos)?

() SIM () NÃO

2) A abordagem prática auxilia a compreensão do assunto?

() SIM () NÃO

3) Você recomendaria a adoção destas ferramentas para o ensino da disciplina de Sistemas Operacionais? Por que?

() SIM () NÃO
