

# Universidade Federal do Paraná

Loirto Alves dos Santos

Luiz Henrique Pires de Camargo

## **Vírus de computador**

Uma abordagem do código polimórfico

Monografia apresentada junto ao curso de Ciência da Computação, do Departamento de Informática, do Setor de Ciências Exatas, como requisito parcial para a obtenção do título de Bacharel.

**Orientador:** Prof. Dr. Bruno Müller Junior

**Curitiba - PR**

2012

*A nossos pais. Sem eles nada disso teria sentido.*

# Agradecimentos

- A Deus
- A nosso esforço e dedicação que, apesar de serem poucos, nos valeram muito.
- Aos professores pela paciência e dedicação

# Resumo

Este trabalho tem por finalidade realizar um estudo sobre alguns algoritmos e técnicas de polimorfismo utilizadas para criar vírus de computador e o quanto elas tornam difícil - e algumas vezes até mesmo impossível - a detecção do código malicioso.

**Palavras-chave:** Vírus, Vírus de computador, Vírus polimórfico, polimorfismo.

# Abstract

This paper aims to conduct a study of some algorithms and techniques used to create polymorphic computer viruses and how they make it difficult - and sometimes even impossible - to detect the malicious code.

**Keywords:** Virus, Computer Virus, Polymorphic virus, Polymorphism.

# Sumário

<b>Agradecimentos</b>	<b>iii</b>
<b>Resumo</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>Sumário</b>	<b>vi</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Revisão bibliográfica</b>	<b>2</b>
2.1 Antivírus . . . . .	2
2.2 História . . . . .	2
2.3 Antivirus e SO . . . . .	2
2.3.1 DOS . . . . .	3
2.3.2 Windows . . . . .	3
2.3.3 Linux . . . . .	3
2.4 Técnicas de detecção . . . . .	4
2.4.1 Virus de pendrive . . . . .	4
2.4.2 Virus de macro . . . . .	4
2.4.3 Virus Polimórficos . . . . .	5

<b>3</b>	<b>Descrição conceitual do trabalho</b>	<b>6</b>
<b>4</b>	<b>Detalhes do trabalho</b>	<b>7</b>
<b>5</b>	<b>Conclusão</b>	<b>8</b>
	<b>Referências Bibliográficas</b>	<b>9</b>
<b>A</b>	<b>Estrutura de Arquivos PE e ELF</b>	<b>16</b>
A.1	Arquivo PE . . . . .	16
A.1.1	Estrutura de arquivo PE. . . . .	16
A.1.2	PE - Cabeçalho . . . . .	17
A.1.3	Tabela de Seções . . . . .	17
A.1.4	Páginas de imagem . . . . .	18
A.1.5	Importação . . . . .	18
A.1.6	Exportação . . . . .	18
A.1.7	Correção . . . . .	19
A.1.8	Recursos . . . . .	19
A.1.9	Debug . . . . .	19
A.2	Arquivo ELF . . . . .	19
A.2.1	A estrutura do arquivo ELF . . . . .	20

# Capítulo 1

## Introdução

Nossa vida moderna é extremamente dependente de computadores: desktops, notebooks, netbooks, PDA, celulares, satélites, veículos, microondas, televisores, gps, bancos, energia elétrica, comunicações ..., enfim, uma gama enorme de exemplos poderiam ser citados. Dentro deste contexto, os vírus de computador (e suas variações) são uma ameaça real à qual todos - direta ou indiretamente - estamos expostos.



## Revisão bibliográfica

### 2.1 Antivírus

Os antivirus são softwares criados para analisar, detectar, eliminar e impedir os virus informáticos ou ao menos diminuir a intensidade do ataque. Foram criados pela necessidade de que os virus impediam a utilização do sistema. Os virus atuais são mais poderosos, e ainda existem outros não tão fortes que são utilizados como piada ou somente para incomodar, se espalhar pelos computadores sem fazer mal à máquina e sim à paciência do usuário.

### 2.2 História

O primeiro antivirus foi criado em 1988 por Denny Yanuar Ramdhani. Era uma vacina ao virus Brain, um virus de boot, além de remover o virus imunizava o sistema contra uma nova infecção. A forma de desinfectar era remover as entradas do virus no pc e já bloqueava ests fraquezas para impedir um novo ataque. Ainda em 1988 um virus foi projetado para infectar com a "ajuda"da BBS, nisto John McAfee, desenvolveu o VirusScan, primeira vacina para o virus.

### 2.3 Antivirus e SO

Por enquanto existe uma dependência dos virus para com os sistemas operacionais, pois afetam o modo em que o executável interage com o sistema, e pedidos especiais são

feitos pelo próprio SO e cada qual o faz de forma diferente, ou seja um vírus que funciona em windows nunca funcionaria em linux, só se fossem chamadas suas APIs, como feito pelo wine no sistema linux, e mesmo assim não teria todo o potencial de infecção, já que é preparado para a estrutura do sistema para o qual foi projetado.

### 2.3.1 DOS

No sistema DOS o anti-virus não funciona em "tempo real", somente como scanner, normalmente era colocado no boot do sistema para varrer o sistema em busca de novas infecções, e outras verificações somente se chamado pelo usuário. Sendo infectado no meio de uma tarefa o vírus já se propagou e danificou diversas areas e somente será percebido na nova execução do antivírus.

### 2.3.2 Windows

Já no windows o antivírus protege as principais formas de ataque, para este sistema. continua a utilizar o scanner, como no DOS. Ganhou a função de monitoramento, com diversas ferramentas para encontrar padrões de vírus. A cada executável aberto há esta verificação, o que compromete o desempenho do computador. A cada periodo pré-determinado há uma varredura sobre os arquivos do sistema para verificar arquivos infectados, remove o vírus e tenta manter a integridade do arquivo. Se encontra um padrão de infecção mas ainda não existe "vacina" para remoção diversos sistemas de proteção utilizam a ferramenta de "quarentena", ou seja mantém o arquivo infectado em um espaço que não pode ser "alcançado" pelo usuário até que possa restaurar o arquivo, ou ao menos conheça o vírus.

### 2.3.3 Linux

Não são muito populares neste sistema. Por enquanto não há uma grande preocupação, nem pela parte de usuários e nem pela parte de desenvolvedores. O que existe hoje são alguns sistemas que detectam vírus para windows pelo linux, para fazer uma manutenção do sistema. E mesmo assim não são tão "potentes" quanto os de windows, não há muita preocupação em desenvolvê-los.

## 2.4 Técnicas de detecção

São diversas as técnicas de detecção dentre elas: Heurística: Que significa descobrir. Estuda o comportamento, estrutura e características para analisar se é perigoso ao sistema ou inofensivo. Emulação: Abre o arquivo em uma virtualização do sistema, e analisa os efeitos sobre o sistema. Arquivo monitorado: Mantém um arquivo no sistema e o monitora, se ele modificar alguma característica é porque o sistema foi infectado. E então o antivírus toma as precauções necessárias. Assinatura do vírus: Com um trecho de código do vírus tem-se sua assinatura, quando tenta detectar o vírus busca-as para analisar se já não existe dentro do banco de dados do antivírus. Temos o falso positivo, o antivírus com base no comportamento do arquivo o considera infectado, o que dificulta para usuários comuns identificarem as anomalias e utilizar com segurança o sistema.

### 2.4.1 Virus de pendrive

No sistema operacional windows eles se utilizam do arquivo autorun.inf para se autoexecutar e infectar a máquina. sua limpeza é simples, existem alguns antivírus que alteram o conteúdo do autorun e tiram a permissão de gravação do arquivo, e alguns usuários criam um diretório com o nome autorun.inf e isso impede de criar o tal arquivo. os vírus em si funcionam de forma interessante, temos por exemplo o conficker q após infectar o pc ele passa a infectar td pendrive q nel for utilizado, assim como enquanto conectado a internet ele baixa diversos outros vírus e com isso acaba com o sistema e arquivos do usuário. sua prevenção é simples e sua remoção é complicada. ou seja se todos fossem informados de como o vírus funciona a prevenção seria óbvia e este tipo de vírus seria obsoleto.

### 2.4.2 Virus de macro

Os vírus de macro são utilizados dentro de, aparentemente, inofensivos arquivos estilo "office" são scripts executados automaticamente para facilitar a visualização dos arquivos e fazer eles executarem o que teriam de executar, os criadores de vírus aproveitam que macros tem poder de execução e infectam os arquivos colocando dentro a macro código malicioso que o usuário previamente nem notará, e após execução do arquivo já estará infectado e infectará outros. A maior praga disso está nas apresentações de slides, como foi muito difundido por e-mails para passar imagens com animações. O vírus se instala dentro destes arquivos e o usuário desconhece que por trás de tudo que está visualizando um vírus acabou de se instalar em sua máquina.

### 2.4.3 Virus Polimórficos

Ainda não existe uma forma eficaz para se detectar este tipo de vírus, eles não tem um padrão a ser identificado. O que se faz é criar um arquivo de vítima e este fica sempre sendo monitorado, mas o bom vírus polimórfico já está residente em memória e faz o sistema "ver" o arquivo como inalterado e com isso não há mais nada a ser feito. seria uma limpeza manual, sem o auxílio de outra máquina seria inviável, enquanto o vírus se infecta o usuário tentaria localizá-lo e deletá-lo uma guerra perdida.

# Capítulo 3

Descrição conceitual do trabalho

# Capítulo 4

## Detalhes do trabalho

# Capítulo 5

## Conclusão

# Referências Bibliográficas

- [1] K. W. Bowyer, K. Chang e P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *Computer Vision and Image Understanding* **101**, 1, 1–15 (2006).
- [2] W. Zhao, R. Chellappa, P. J. Phillips e A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys* **35**, 4, 399–458 (2003).
- [3] R.-L. Hsu, M. A. Mottaleb e A. K. Jain. Face Detection in Color Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 5, 696–706 (2002).
- [4] M. Turk e A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* **3**, 1, 71–86 (1991).
- [5] P. Viola e M. J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision* **57**, 2, 137–154 (2004).
- [6] M.-H. Yang, D. J. Kriegman e N. Ahuja. Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 1, 34–58 (2002).
- [7] K. I. Chang, K. W. Bowyer e P. J. Flynn. Multiple Nose Region Matching for 3D Face Recognition under Varying Facial Expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**, 10, 1695–1700 (2006).
- [8] X. Lu e A. K. Jain. Multimodal Facial Feature Extraction for Automatic 3D Face Recognition. Technical Report, Department of Computer Science, Michigan State University (2005).
- [9] Y. Wang, J. Liu e X. Tang. Robust 3D Face Recognition by Local Shape Difference Boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 10, 1858–1870 (2010).



- [10] X. Lu e A. K. Jain. Automatic Feature Extraction for Multiview 3D Face Recognition. Em *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, páginas 585–590 (2006).
- [11] M. Pamplona Segundo, L. Silva, O. R. P. Bellon e C. C. Queirolo. Automatic face segmentation and facial landmark detection in range images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **40**, 5, 1319–1330 (2010).
- [12] F. Tsalakanidou, S. Malassiotis e M. G. Strintzis. Face localization and authentication using color and depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 2, 152–168 (2005).
- [13] A. Colombo, C. Cusano e R. Schettini. 3D face detection using curvature analysis. *Pattern Recognition* **39**, 3, 444–455 (2006).
- [14] T. Faltemier, K. W. Bowyer e P. J. Flynn. Using a Multi-Instance Enrollment Representation to Improve 3D Face Recognition. Em *Proc. of the IEEE International Conference on Biometrics: Theory, Applications, and Systems*, páginas 1–6 (2007).
- [15] I. Kakadiaris, G. Passalis, G. Toderici, M. Murtuza, Y. Lu, N. Karampatziakis e T. Theoharis. Three-Dimensional Face Recognition in the Presence of Facial Expressions: An Annotated Deformable Model Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**, 4, 640–649 (2007).
- [16] A. Mian, M. Bennamoun e R. Owens. An Efficient Multimodal 2D-3D Hybrid Approach to Automatic Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**, 11, 1927–1943 (2007).
- [17] R. Lienhart e J. Maydt. An extended set of Haar-like features for rapid object detection. Em *Proceedings of the International Conference on Image Processing*, volume 1, páginas 900–903 (2002).
- [18] J. Fischer, D. Seitz e A. Verl. Face Detection using 3-D Time-of-Flight and Colour Cameras. Em *Proceedings of the 41st International Symposium on Robotics and 6th German Conference on Robotics* (2010).
- [19] M. Böhme, M. Haker, K. Riemer, T. Martinetz e E. Barth. Face Detection Using a Time-of-Flight Camera. *Lecture Notes in Computer Science* **5742**, 167–176 (2009).
- [20] Y. Freund e R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Em *Proceedings of the Second European Conference on Computational Learning Theory*, páginas 23–37 (1995).

- [21] L. Yin, X. Wei, Y. Sun, J. Wang e M. J. Rosato. A 3D Facial Expression Database For Facial Behavior Research. Em *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, páginas 211–216 (2006).
- [22] H. Tang e T. S. Huang. 3D facial expression recognition based on properties of line segments connecting facial feature points. Em *8th IEEE International Conference on Automatic Face and Gesture Recognition*, páginas 1–6 (2008).
- [23] H. Ghorayeb, B. Steux e C. Laurgeau. Boosted Algorithms for Visual Object Detection on Graphics Processing Units. Em *Computer Vision – ACCV 2006*, volume 3852 de *Lecture Notes in Computer Science*, páginas 254–263 (2006).
- [24] X. Lu e A. K. Jain. Automatic Feature Extraction for Multiview 3D Face Recognition. Em *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, páginas 585–590 (2006).
- [25] H. Tang e T. S. Huang. 3D facial expression recognition based on automatically selected features. Em *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, páginas 23–28 (2008).
- [26] I. Mpiperis, S. Malassiotis e M. G. Strintzis. Bilinear Models for 3-D Face and Facial Expression Recognition. *IEEE Transactions on Information Forensics and Security* **3**, 3, 498–511 (2008).
- [27] O. R. P. Bellon e L. Silva. New improvements to range image segmentation by edge detection. *IEEE Signal Processing Letters* **9**, 2, 43–45 (2002).
- [28] O. R. P. Bellon, L. Silva e C. C. Queirolo. 3D face matching using the Surface Interpenetration Measure. Em *Lecture Notes in Computer Science*, volume 3617, páginas 1051–1058 (Springer-Verlag, 2005).
- [29] P. F. U. Gotardo, O. R. P. Bellon, K. L. Boyer e L. Silva. Range image segmentation into planar and quadric surfaces using an improved robust estimator and genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **34**, 6, 2303–2316 (2004).
- [30] B. Gökberk, H. Dutagaci, A. Ulas, L. Akarun e B. Sankur. Representation Plurality and Fusion for 3-D Face Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **38**, 1, 155–173 (2008).

- [31] S. Berreti, A. Del Bimbo, P. Pala e F. J. S. Mata. Geodesic distances for 3D-3D and 2D-3D face recognition. Em *Proc. of the IEEE International Conference on Multimedia and Expo*, páginas 1515–1518 (2007).
- [32] P. J. Besl. *Surface in Range Images Understanding* (Springer-Verlag, 1988).
- [33] P. J. Besl e R. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**, 167–192 (1988).
- [34] P. J. Besl e H. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 2, 239–256 (1992).
- [35] A. M. Bronstein, M. M. Bronstein e R. Kimmel. Three-Dimensional Face Recognition. *International Journal of Computer Vision* **64**, 1, 5–30 (2005).
- [36] J. Y. Cartoux, J. T. Lapreste e M. Richetin. Face authentication or recognition by profile extraction from range images. Em *Proc. of the IEEE Computer Society Workshop on Interpretation of 3D Scenes*, páginas 194–199 (1989).
- [37] K. I. Chang, K. W. Bowyer e P. J. Flynn. Adaptive Rigid Multi-region Selection for Handling Expression Variation in 3D Face Recognition. Em *Proc. of the IEEE Workshop FRGC* (2005).
- [38] K. I. Chang, K. W. Bowyer e P. J. Flynn. Multiple Nose Region Matching for 3D Face Recognition under Varying Facial Expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**, 10, 1695–1700 (2006).
- [39] K. I. Chang, K. W. Bowyer e P. J. Flynn. Effects on facial expression in 3D face recognition. Em *Proc. of the SPIE - Biometric Technology for Human Identification*, volume 5779, páginas 132–143 (2005).
- [40] C. S. Chua e R. Jarvis. Point Signatures: A New Representation for 3D Object Recognition. *International Journal of Computer Vision* **25**, 1, 63–85 (1997).
- [41] D. Colbry, G. Stockman e A. K. Jain. Detection of Anchor Points for 3D Face Verification. Em *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, página 118 (2005).
- [42] T. Faltemier, K. W. Bowyer e P. J. Flynn. Rotated Profile Signatures for robust 3D feature detection. Em *Proc. of the 8th IEEE International Conference on Automatic Face and Gesture Recognition*, páginas 1–7 (2008).

- [43] P. F. Felzenszwalb e D. P. Huttenlocher. Distance Transforms of Sampled Functions. Technical Report, Cornell Computing and Information Science (2004).
- [44] R. C. Gonzalez e R. E. Woods. Digital Image Processing (Addison-Wesley, 1992).
- [45] C. Heshner, A. Srivastava e G. Erlebacher. Principal component analysis of range images for facial recognition. Em *Proc. of the International Conference on Imaging Science, Systems and Technology* (2002).
- [46] A. K. Jain e R. C. Dubes. Algorithms for clustering data (Prentice-Hall, 1988).
- [47] S. Kirkpatrick, C. D. Gelatt e M. P. Vecchi. Optimization by Simulated Annealing. *Science* **220**, 4598, 671–680 (1983).
- [48] J. Kittler, M. Hatef, R. P. W. Duin e J. Matas. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 3, 226–239 (1998).
- [49] X. Lu, A. K. Jain e D. Colbry. Matching 2.5D Face Scans to 3D Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**, 1, 31–43 (2006).
- [50] J. Macqueen. Some methods for classification and analysis of multivariate observations. Em *Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, páginas 281–297 (1967).
- [51] A. B. Moreno, A. Sánchez, J. F. Vélez e F. J. Díaz. Face recognition using 3d surface-extracted descriptor. Em *Proc. of the Irish Machine Vision and Image Processing* (2003).
- [52] G. Pan, Y. Wu e Z. Wu. Investigating Profile Extracted from Range Data for 3D Face Recognition. Em *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, páginas 1396–1399 (2003).
- [53] C. C. Queirolo, L. Silva, O. R. P. Bellon e M. Pamplona Segundo. 3D Face Recognition using Simulated Annealing and the Surface Interpenetration Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (forthcoming papers) (2009).
- [54] M. Romero-Huertas e N. Pears. 3D Facial Landmark Localisation by Matching Simple Descriptors. Em *Proc. of the 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*, páginas 1–6 (2008).

- [55] M. Pamplona Segundo, C. C. Queirolo, O. R. P. Bellon e L. Silva. Automatic 3D facial segmentation and landmark detection. *Proc. of the 14th International Conference on Image Analysis and Processing* páginas 431–436 (2007).
- [56] L. G. Shapiro e G. C. Stockman. *Computer Vision* (Prentice-Hall, 2001).
- [57] L. Silva, O. R. P. Bellon e K. L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 5, 762–776 (2005).
- [58] K. Sobottka e I. Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing-Image Communication* **12**, 3, 263–281 (1998).
- [59] P. Torr e A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* **78**, 138–156 (2000).
- [60] T.-H. Yu e Y.-S. Moon. A Novel Genetic Algorithm for 3D Facial Landmark Localization. Em *Proc. of the 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*, páginas 1–6 (2008).
- [61] J. L. Hennessy e D. A. Patterson. *Computer Architecture: A Quantitative Approach* (Morgan Kaufmann, 2007).
- [62] G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. *Spring Joint Computer Conf.* páginas 483–485 (1967).
- [63] L. Lamport. How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs. *IEEE Transactions on Computers* **C-28**, 690–691 (1979).
- [64] K. Fatahalian e M. Houston. A Closer Look at GPUs. Em *Communications of the ACM*, volume 51, páginas 50–57 (2008).
- [65] T. Braunl e S. Feyrer. *Parallel Image Processing* (Springer, 2001).
- [66] M. Flynn. Very high-speed computing systems. *Proceedings of the IEEE* **54**, 12, 1901 – 1909 (1966). doi: 10.1109/PROC.1966.5273.
- [67] M. Pamplona Segundo, O. R. P. Bellon e L. Silva. Real-Time Scale-Invariant Face Detection on Range Images. *IEEE International Conference on System, Man, and Cybernetics (Aceito)* (2011).

# Appendices

# Estrutura de Arquivos PE e ELF

## A.1 Arquivo PE

O formato de arquivo PE (Portable Executable Format File) é o último utilizado para plataforma Microsoft.

### A.1.1 Estrutura de arquivo PE.

DOS 2 - Cabeçalho EXE compatível	Seção DOS 2.0 (para compatibilidade com DOS somente)
Não utilizado	
OEM - Identificador OEM - Info Offset para cabeçalho PE	
DOS 2.0 Stub Program & Reloc. Table	
Não utilizado	
PE - Cabeçalho	Palavras limitadas a 8 bytes
Tabela de seções	
Image Pages · Info de Importação · Info de Exportação · Info de correção · Info de recursos · Info de debug	

### A.1.2 PE - Cabeçalho

Temos no cabeçalho uma estrutura dividida em campos com palavras de 4 bytes, enfatizamos alguns deles abaixo:

Tipo de CPU: o campo informa qual o tipo de CPU para a qual o executável foi projetado.

Número de Seções: o campo informa o número de entradas na tabela de seções.

Marca de Tempo/Data: Armazena a data de criação ou modificação do arquivo.

Flags: Bits para informar qual o tipo de arquivo ou quando há erros em sua estrutura.

LMAJOR/LMINOR: maior e menor versão do linkador para o executável.

Seção de alinhamento: O valor de alinhamento das seções. Deve ser múltiplo de 2 entre 512 e 256M. O valor padrão é 64K.

OS MAJOR/MINOR = Versões limitantes (maior e menor) do sistema operacional.

Tamanho da Imagem: Tamanho virtual da imagem, contando todos os cabeçalhos. E o tamanho total deve ser múltiplo da seção de alinhamento.

Tamanho do Cabeçalho: Tamanho total do cabeçalho. O tamanho combinado de cabeçalho do DOS, cabeçalho do PE e a tabela de seções.

FILE CHECKSUM: Checksum do arquivo em si, é setado como 0 pelo linkador.

Flags de DLL: Indica qual o tipo de leitura que deve ser feita, processos de inicialização e terminação de leitura e de threads.

Tamanho reservado da pilha: tamanho de pilha reservado ao programa, o valor real é o valor efetivo, se o valor reservado não tiver no sistema ele será paginado.

Tamanho efetivo da pilha: tamanho efetivo.

Tamanho Reservado da HEAP: Tamanho reservado a HEAP.

Tamanho efetivo da HEAP: Valor efetivo para a HEAP.

### A.1.3 Tabela de Seções

O número de entradas da tabela de seções é dado pelo campo de número de seções que está no cabeçalho. As entradas se iniciam em 1. Segue imediatamente o cabeçalho do PE. A ordem de dados e memória é selecionado pelo ligador. Os endereços virtuais



para as seções são confirmados pelo ligador de forma crescente e adjacente, e devem ser múltiplos da Seção de alinhamento, que também é fornecida no cabeçalho do PE. Abaixo alguns de uma seção nesta tabela, divididos em palavras de 8 bytes:

Nome da Seção: Campo com 8 bytes nulos para representar o nome da seção em ASCII.

Tamanho virtual: O tamanho virtual é o alocado quando a seção é lida.

Tamanho físico: O tamanho de dados inicializado no arquivo para a seção. É múltiplo do campo de alinhamento do arquivo do cabeçalho do PE e deve ser menor ou igual ao tamanho virtual.

Offset físico: Offset para apontar a primeira página da seção. É relativo ao início do arquivo executável.

Flags da seção: Flags para sinalizar se a seção é de código, se está inicializada ou não, se deve ser armazenada, compartilhada, paginável, de leitura ou para escrita.

#### A.1.4 Páginas de imagem

A página de imagens contém todos os dados inicializados e todas as seções. As seções são ordenadas pelo endereço virtual reservado a elas. O Offset que aponta para a primeira página é especificado na tabela de seções como visto na subseção acima. Cada seção inicia com um múltiplo da seção de alinhamento.

#### A.1.5 Importação

A informação de importação inicia com uma tabela de diretórios de importação que descreve a parte principal da informação de importação. A tabela de diretórios de importação contém informação de endereços que são utilizados nas referências de correção para pontos de entrada com uma DLL. A tabela de diretórios de importação consiste de um vetor de entradas de diretórios, uma entrada para cada referência a DLL. A última entrada é nula o que indica o fim da tabela de diretórios.

#### A.1.6 Exportação

A informação de exportação inicia com a tabela de diretórios de exportação que descreve a parte principal da informação de exportação. A tabela de diretórios de exportação

contém informação de endereços que são utilizados nas referencias de correção para os pontos de entrada desta imagem.

### A.1.7 Correção

A tabela de correção contém todas as entradas de correção da imagem. O tamanho total de dados de correção no cabeçalho é o número de bytes na tabela de correção. A tabela de correção é dividida em blocos de correção. Cada bloco representa as correções para um página de 4K bytes. Correções que são resolvidados pelo ligador necessitam ser processadas pelo carregador, a menos que a imagem não possa ser carregada na Base de imagens especificada no cabeçalho do PE.

### A.1.8 Recursos

Recursos são indexados por uma arvore binária ordenada. O design como um todo pode chegar a  $2^{31}$  níveis, entretanto, NT utiliza somente 3 níveis: o mais alto com o *tipo*, no subsequente *nome*, depois a *língua*.

### A.1.9 Debug

A informação de debug é definido por um debugador que não é controlado pelo PE ou pelo ligador. Somente é definido pelo PE os dados da tabela de diretório de debug.

## A.2 Arquivo ELF

Explicação

### A.2.1 A estrutura do arquivo ELF

Arquivo Realocável	Arquivo Carregável
Cabeçalho ELF	Cabeçalho ELF
Tabela do cabeçalho do programa (opcional)	Tabela do cabeçalho do programa
seção 1	Segmento 1
seção 2	
...	Segmento 2
seção n	...
Tabela de cabeçalho de seção	Tabela de cabeçalho de seção(opcional)