

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
CURSO CIÊNCIAS DA COMPUTAÇÃO

Antônio Augusto Bandeira de Oliveira

Gabriel Antony Olsson Schlagenhauser

Igor Terplak Gutierrez

Jonas Oliveira de Assis

SISTEMAS OPERACIONAIS:

RASPBIAN

CURITIBA

2023

Antônio Augusto Bandeira de Oliveira

Gabriel Antony Olsson Schlagenhauser

Igor Terplak Gutierrez

Jonas Oliveira de Assis

SISTEMAS OPERACIONAIS:

RASPBIAN

Trabalho acadêmico apresentado ao curso de Bacharelado em Ciências da Computação, da Pontifícia Universidade Católica do Paraná, como requisito à obtenção de nota parcial na disciplina de Fundamentos de Sistemas Ciberfísicos.

Orientador: Prof. Andrey Cabral Meira

CURITIBA

2023

SUMÁRIO

1. INTRODUÇÃO	4
2. SISTEMAS OPERACIONAIS	5
2.1. ABSTRAÇÃO E GERÊNCIA DE RECURSOS	5
2.2. FUNCIONALIDADES	6
2.3. CATEGORIAS	7
2.4. INTERRUPÇÕES E EXCEÇÕES.....	8
2. RASPBIAN	9
2.1. DEFINIÇÃO E FUNCIONALIDADE	9
2.2. SISTEMAS DE ARQUIVOS	9
2.3. REQUISITOS DE SISTEMA	10
2.4. VANTAGENS E DESVANTAGENS	10
2.5. CURIOSIDADES.....	10
3. CONSIDERAÇÕES FINAIS	11
4. REFERÊNCIAS BIBLIOGRÁFICAS	11

1. INTRODUÇÃO

Um computador sem programas possui tanta funcionalidade quanto um relógio que não marca as horas, ou seja, possui uma funcionalidade limitada, tal qual um peso de papel.

Os softwares são responsáveis pelo armazenamento, processamento e recuperação de informações. Em suma, todas as benesses que facilitam o cotidiano tanto destes autores, quanto de ti, querido leitor, são graças aos softwares.

Tanenbaum e Woodhull (2008) dividem os softwares em dois tipos: programas de sistemas, o qual é responsável por gerenciar a operação do computador em si, e programas aplicativos, os quais realizam o trabalho real desejado pelo usuário.

Os sistemas operacionais, mote deste trabalho acadêmico, são classificados como os programas de sistemas mais básicos, sendo responsáveis pelo controle dos recursos do computador e fornecimento de uma base sobre a qual os programas aplicativos podem operar.

Portanto, podemos, desde já, afirmar que o objetivo-mor dos sistemas operacionais é tornar o computador conveniente ao uso, transformando de uma pilha de metal, chips e micro condutores para uma máquina capaz de gerenciar desde pequenos cálculos até países inteiros.

O presente trabalho acadêmico apresentará uma breve concepção dos objetivos de um sistema operacional e se aprofundará num sistema operacional específico, o Raspbian. Permitindo ao leitor compreender o objetivo do sistema supramencionado, suas origens, funcionalidades, requisitos e demais assuntos.

2. SISTEMAS OPERACIONAIS

2.1. ABSTRAÇÃO E GERÊNCIA DE RECURSOS

Conforme mencionado anteriormente, os sistemas operacionais são os responsáveis pela integração entre aplicativos e hardware, contudo, tal tarefa é extremamente complexa, haja vista que cada dispositivo físico possui uma característica única e uma interface própria.

Tais complexas tarefas são executadas várias vezes por minuto, portanto, o sistema operacional deve definir interfaces abstratas para os recursos do hardware.

Carlos Maziero (2019) afirma que tal abstração visa os seguintes objetivos:

“Prover interfaces de acesso aos dispositivos, mais simples de usar que as interfaces de baixo nível, para simplificar a construção de programas aplicativos(...)

Tornar os aplicativos independentes do hardware(...)

Definir interfaces de acesso homogêneas para dispositivos com tecnologias distintas. Através de suas abstrações, o sistema operacional permite aos aplicativos usar a mesma interface para dispositivos diversos. Por exemplo, um aplicativo acessa dados em disco através de arquivos e diretórios, sem precisar se preocupar com a estrutura real de armazenamento dos dados, que podem estar em um disquete, um disco SATA, uma máquina fotográfica digital conectada à porta USB, um CD ou mesmo um disco remoto, compartilhado através da rede”.

Contudo, na interação de hardware e software, intermediada pelo sistema operacional, podem ocorrer inúmeros erros, quando, por exemplo, dois ou mais aplicativos precisam dos mesmos recursos para poder executar. Em casos como esse, o sistema operacional deve estabelecer diretrizes para gerenciar o uso dos recursos do hardware pelos aplicativos.

O autor Carlos Maziero (2019) traz alguns exemplos onde a gerência de recursos se faz necessária, vejamos:

“Cada computador normalmente possui menos processadores que o número de tarefas em execução. Por isso, o uso desses processadores deve ser distribuído entre os aplicativos presentes no sistema, de forma que cada um deles possa executar na velocidade adequada para cumprir suas funções sem prejudicar os demais. O mesmo ocorre com

a memória RAM, que deve ser distribuída de forma justa entre as aplicações.

(...)

Ataques de negação de serviços (DoS – *Denial of service*) são comuns na Internet. Eles consistem em usar diversas técnicas para forçar um servidor de rede a dedicar seus recursos para atender um determinado usuário, em detrimento dos demais. Por exemplo, ao abrir milhares de conexões simultâneas e um servidor de e-mail, um atacante pode reservar para si todos os recursos do servidor, fazendo com que os demais usuários não sejam mais atendidos. É responsabilidade do sistema operacional do servidor detectar tais situações e impedir que todos os recursos do sistema sejam monopolizados por um só usuário.”

Portanto, um sistema operacional tem como objetivo o gerenciamento de recursos do hardware permitindo que o usuário consiga rodar todas suas aplicações com conflitos de uso mínimos. Tudo isso através da abstração e gerenciamento de recursos.

2.2. FUNCIONALIDADES:

Conforme elencado por Mazieiro (2019) o sistema operacional deve atuar em diversas frentes para cumprir com seus objetivos de abstração e gerência. A fim de cumprir com seus objetivos os sistemas operacionais contam com funcionalidades como gerência do processador, gerência de memória, dispositivos, arquivos e proteção.

Contudo, as gerências supra elencadas fazem parte do “pacote básico” de um sistema operacional, ou seja, cada sistema operacional disponível conta com ainda mais funcionalidades, únicas em alguns casos, que vão desde interface gráfica, à fontes de energia.

Outrossim, Maziero, citando Levin, afirma que na construção de um sistema operacional é importante a separação entre conceitos de política e mecanismo. Sendo política os aspectos de decisão mais abstratos, que podem ser resolvidos por algoritmos de nível mais alto, como, por exemplo, a distribuição de memória para cada aplicativo.

Enquanto mecanismo se consideram os procedimentos de baixo nível usados para implementar as políticas, ou seja, para atribuir ou retirar memória de uma aplicação. Devendo estes serem suficientemente genéricas para suportar mudanças de política sem necessidade de modificações.

2.3. CATEGORIAS:

Sistemas Operacionais, como um grupo, podem ser classificados de diversas formas: tamanho de código, suporte a recursos específicos, etc.

Maziero (2019) elenca alguns tipos de sistemas operacionais, quais sejam, *in verbis*:

“Batch (de lote): os sistemas operacionais mais antigos trabalhavam “por lote”, ou seja, todos os programas a executar eram colocados em uma fila, com seus dados e demais informação para a execução. (...) Atualmente, este conceito se aplica a sistemas que processam tarefas sem interação direta com os usuários, como os sistemas de processamento de transações bancárias. (...) Exemplos clássicos desses sistemas incluem o IBM OS/360 e o VAX/VMS, entre outros.

De rede: um sistema operacional de rede deve possuir suporte à operação em rede, ou seja, a capacidade de oferecer às aplicações locais recursos que estejam localizados em outros computadores da rede, como arquivos e impressoras. (...) A maioria dos sistemas operacionais atuais oferece esse tipo de funcionalidade.

Distribuído: em um sistema operacional distribuído, os recursos de cada computador estão disponíveis a todos na rede, de forma transparente aos usuários. (...) Em uma aplicação na nuvem, o usuário interage com a interface da aplicação em um computador ou telefone, mas não tem uma visão clara das máquinas onde seus dados estão sendo processados e armazenados.

Multiusuário: um sistema operacional multiusuário deve suportar a identificação do “dono” de cada recurso dentro do sistema (arquivos, processos, áreas de memória, conexões de rede) e impor regras de controle de acesso para impedir o uso desses recursos por usuários não autorizados. Essa funcionalidade é fundamental para a segurança dos sistemas operacionais de rede e distribuídos. Grande parte dos sistemas atuais são multiusuários.

Servidor: um sistema operacional servidor deve permitir a gestão eficiente de grandes quantidades de recursos (...). Normalmente um sistema operacional servidor também tem suporte a rede e multiusuários.

Desktop: um sistema operacional “de mesa” é voltado ao atendimento do usuário doméstico e corporativo para a realização de atividades corriqueiras, como edição de textos e gráficos, navegação na Internet e reprodução de mídia. (...) Exemplos de sistemas desktop são os vários sistemas Windows (XP, Vista, 7, 10, etc.), MacOS e Linux.

Móvel: um sistema operacional móvel é usado em equipamentos de uso pessoal compactos como *smartphones* e *tablets*. Nesse contexto, as principais prioridades são a gestão eficiente da energia (bateria), a conectividade nos diversos tipos de rede (wifi, GSM, Bluetooth, NFC, etc) e a interação com uma grande variedade de sensores (GPS, giroscópio, luminosidade, tela de toque, leitor de digitais, etc). Android e iOS são bons exemplos desta categoria.

Embarcado: (...) construído para operar sobre um hardware com poucos recursos de processamento, armazenamento e energia. (...) Muitas vezes um sistema operacional embarcado se apresenta na forma de uma biblioteca a ser ligada ao programa da aplicação durante sua compilação. LynxOS, TinyOS, Contiki e VxWorks são exemplos de sistemas operacionais embarcados.

Tempo real: são sistemas nos quais o tempo é essencial. (...) A estrutura interna de um sistema operacional de tempo real deve ser construída de forma a minimizar esperas e latências imprevisíveis, como tempos de acesso a disco e sincronizações excessivas. Exemplos de sistemas operacionais de tempo real incluem o QNX, RT-Linux e VxWorks. Muitos sistemas embarcados têm características de tempo real, e vice-versa.

Portanto, como podemos observar, enquanto inúmeros forem os sistemas operacionais existentes, inúmeras serão as categorias de classificação. Cabendo ao usuário escolher o que mais se adapta às suas necessidades.

2.4. INTERRUPÇÕES E EXCEÇÕES

Com intuito de preservar alocação de memória e reduzir o consumo energético, bem como para controlar entrada e saída de dados, não é proveitoso que a CPU tenha que ficar monitorando os status de dispositivos. Portanto, para tal, o mecanismo de interrupção permite que o hardware alerte o CPU quando há algo a ser feito.

Ou seja, interrupções e exceções são eventos que interrompem o fluxo normal de execução da instrução. Sendo a interrupção um evento inesperado de fora do processo, e a exceção um evento inesperado de dentro do processador.

Toda vez que ocorre um destes eventos, o hardware começa a executar o código que executa uma ação em resposta à exceção ou interrupção. Tal código é chamado de tratador de interrupção.

3. RASPBIAN

3.1. DEFINIÇÃO E FUNCIONALIDADE:

Após essa breve introdução sobre as funcionalidades básicas inerentes à quase todos os sistemas operacionais, passamos à análise do objeto de estudo deste presente trabalho: O sistema operacional Raspbian.

Em descrição fornecida pela própria desenvolvedora o Raspbian é um sistema operacional gratuito, baseado no Debian¹, otimizado para Raspberry Pi² que está em constante atualização.

Contudo, trata-se de um sistema operacional completo, contando com ferramentas de acesso à rede com o navegador Midori, aplicativos de criação de texto, planilhas e apresentação em slides com o libreoffice, dentre outros.

Ou seja, a funcionalidade do Raspbian é fornecer um ambiente completo para os usuários de Raspberry Pi, permitindo ao usuário usufruir ao máximo toda o potencial do computador de placa.

3.2. SISTEMAS DE ARQUIVOS;

Por se tratar de um sistema operacional baseado em Linux, o Raspbian utiliza como sistema de arquivos padrão o ext4 (Extended File System 4) o que possibilita o armazenamento de arquivos de grande porte sem prejudicar o desempenho do sistema.

Contudo, vale ressaltar que o Raspbian, por se tratar de um sistema operacional de código aberto, também pode ser configurado para usar outros sistemas de arquivos, como FAT32 (File Allocation Table 32) ou NTFS (New Technology File System).

¹ Debian é um sistema operacional de código aberto, também gratuito, que utiliza o Linux em conjunto com outros aplicativos que juntos compõem um sistema operacional funcional e muito bem aceito pelos seus usuários.

² Raspberry Pi, em suma, é um computador em formato de placa, que permite a integração com periféricos, bem como comporta inúmeros sistemas operacionais.

3.3. REQUISITOS DE SISTEMA:

É necessário ter em mente que os requisitos de sistema variam de acordo com a versão do raspbian a ser instalada, bem como vale ressaltar que se trata de um sistema operacional desenvolvido para ser rodado num computador de placa única, o Raspberry Pi.

Portanto, os requisitos de sistema são muito modestos, podendo ser encontrado na documentação do sistema operacional requisitos mínimos de 512MB de memória RAM, CPU de 700Mhz, e um cartão de memória de 4Gb.

Contudo, um armazenamento superior é recomendado caso o usuário deseje usufruir da vasta biblioteca de aplicativos do sistema operacional.

3.4. VANTAGENS E DESVANTAGENS

O Raspbian é um sistema operacional deveras interessante, contudo, como já mencionado anteriormente, ele é baseado no Debian, ou seja, Linux, portanto, isso pode vir a assustar os usuários mais casuais, haja vista se tratar de um sistema operacional um pouco mais complexo de se lidar.

Contudo, se ignorarmos a curva de aprendizado do Raspbian, temos em nossas mãos um sistema operacional gratuito extremamente eficiente, de fácil instalação, com uma vasta biblioteca de aplicativos e super leve, capaz de rodar desde os computadores mais modestos até nas máquinas mais potentes.

3.5. CURIOSIDADES

Apesar do nome, o Raspbian não é afiliado à *Raspberry Pi Foundation*, empresa criadora do microcomputador. O sistema operacional em voga foi criado por uma equipe pequena de desenvolvedores que são grandes fãs do Raspberry Pi e do Projeto Debian.

Por se tratar de um sistema operacional de código aberto, constantemente surgem versões extraoficiais do Raspbian, as mais famosas são Adafruit e RaspRazor

4. CONSIDERAÇÕES FINAIS

Conforme exposto, sistemas operacionais são os responsáveis por gerenciar aplicativos e otimizar o uso de hardware, seja com alocação de memória, consumo de energia, etc. Bem como podem ser catalogados de infinitas formas, a depender da do critério de escolha do usuário.

Dentre uma infinidade de sistemas operacionais disponíveis no mercado, este trabalho acadêmico se focou no sistema operacional Raspbian, um sistema leve, gratuito e de código aberto, criado por desenvolvedores apaixonados pelo computador de placa única Raspberry Pi.

O passar do tempo não traz apenas melhorias em hardwares, mas também nos softwares que neles operam. Tal evolução pode ser vista ao verificarmos a evolução do sistema objeto de estudo desde trabalho acadêmico – Raspbian – o qual, com o passar dos anos, fora se tornando mais otimizado com o intuito de extrair o máximo do Raspberry Pi.

5. REFERÊNCIAS BIBLIOGRÁFICAS

MAZIEIRO, Carlos. Capítulo 1. *In*: MAZIEIRO, Carlos. **Sistemas Operacionais**: Conceitos e Mecanismos. [S. l.: s. n.], 2019. p. 2-12. Disponível em: <https://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-01.pdf>. Acesso em: 26 maio 2023.

TANENBAUM, Andrew S.; WOODHULL, Albert S. **Sistemas Operacionais**: Projeto e Implementação. Terceira Edição. ed. Porto Alegre: Bookman, 2008. ISBN 85-7164-555-8.