



Introdução a Minix

Projeto 1

Resumo

Este projeto é a introdução básica a **Minix**. Devem baixar e instalar Minix num emulador¹, recompilar o kernel, e fazer uma mudança simples para adicionar capacidades de emulação a Minix.

1 Instruções

1.1 Trabalho a fazer

1. Este projeto se desenvolverá em trios. Devem informar sua equipe (no link <https://goo.gl/forms/okdR1A7EV3nD1l162>) para que o professor possa criar seu repositório no grupo da disciplina no **Gitlab IC**.
2. Leiam detalhadamente as instruções a realizar (pode que precisem mais de uma leitura para entender o trabalho).
3. Precisarão mais informação que os enlaces deste enunciado. Procurem informação enquanto resolvem os problemas. E ainda mais importante, documentem-na para explicá-la no seu relatório.
4. Devem seguir a guia apresentada na Seção 2 para a instalação e configuração de seu sistema.
5. Devem criar uma rama **projeto-1** no seu repositório dentro do grupo **mc504/2018s1** do Gitlab do IC para manter os resultados do seu trabalho. Atualmente, os pipelines para a integração contínua não estão funcionando. Por tanto, devem de criar a rama só para manter suas mudanças no kernel. Se espera que pipeline este funcional no final da primeira semana para que possam testar suas mudanças num ambiente neutro. Fiquem atentos aos avisos no website da disciplina para mais informação sobre a integração continua de seu projeto.
6. Devem responder as questões apresentadas na Seção 3 no seu relatório.

1.2 Entregáveis

Devem entregar no Classroom

¹Os pipelines de integração continua e de avaliação usam **Qemu**, mas no desenvolvimento local vocês podem usar o que preferiam.

- **relatorio.tex**: o código fonte para gerar seu relatório, e
- **relatorio.pdf**: o arquivo de seu relatório.

Caso o código fonte do seu relatório tenha vários arquivos (os arquivos criados pelo **L^AT_EX** não devem ser inclusos) devem empacotar eles e subir um arquivo só.

1.3 Data entrega

A data de entrega é no dia 27 de março. Devem subir os arquivos no Classroom dentro da tarefa do projeto. Não precisam duplicar as submissões. Uma pessoa por grupo deverá submeter o projeto.

Se recomenda submeter os trabalhos com antecedência e não esperar aos últimos minutos para a submissão. Não se considerarão correios nem commits enviados depois da data limite (independentemente do motivo do atraso).

2 Instalação de Minix

Devem instalar a última versão de Minix (3.4.0). Baixem a última imagem ISO do **website do projeto** (R3.4.0). Devem decidir onde desejam instalar Minix, ou em hardware real ou numa máquina virtual, como **Qemu**, **Virtual Box**, **Bochs**, ou **VMWare**. Notem que uma instalação nativa pode executar-se mais rápido. Porém, numa máquina virtual têm a vantagem de reiniciar, pausar, o apagar a máquina facilmente. Além disso, não têm que se preocupar de que a instabilidade no seu código comprometa a máquina onde estão trabalhando.

Têm que usar um versionador de código para administrar suas entregas. Não seria bom que o trabalho de todo um dia se perca quando a máquina virtual quede inacessível, e seu código está no disco duro que não pode ser mais acessado. Por isso, devem usar um repositório no grupo **mc504/2018** da disciplina deste semestre no Gitlab do IC (<https://gitlab.ic.unicamp.br/mc504/2018-s1>) —o repositório será criado uma vez submetam os nomes da equipe. Neste repositório devem clonar ou fazer um fork da rama **master** do repositório Minix que sera usado durante a disciplina (<https://gitlab.>

ic.unicamp.br/mc504/minix). Este repositório não é mais que um fork do código original do Minix. Porém, se o repositório tiver problemas ou mudanças, elas serão resolvidas nessa versão.

Existe documentação sobre a instalação de Minix sobre hardware, por exemplo vejam [este enlace](#). Para a instalação em máquinas virtuais podem seguir [estas instruções](#). Em general, a instalação envolve os seguintes passos (independentemente do médio que escolham):

- particionar o disco duro,
- *bootear* Minix desde a imagem que baixaram,
- executar o script de início, e
- reiniciar a máquina desde a instalação de Minix.

2.1 Configurando Minix

Com Minix instalado, podem fazer algumas coisas para melhorar a usabilidade do sistema (ver <http://wiki.minix3.org/doku.php?id=usersguide:postinstallation>). A primeira é fazer que a rede funcione. Na versão 3.4.0 isto deveria funcionar depois da instalação. Caso não funcione têm que revisar a documentação para poder ter uma rede funcionando.

Com a rede ativa, podem instalar outros aplicativos que ajudem no uso do sistema. Por exemplo, um editor de texto em consola (como *vim*), *openssh* e *rsync* (para intercambiar arquivos entre Minix e seu sistema), ou uma nova shell (por exemplo, *bash*). Notem que alguns destes aplicativos podem estar instaladas com a imagem que baixaram. Adicionalmente, devem de colocar o código de Minix em */usr/src* e devem familiarizar-se com os elementos que se encontram nas diferentes pastas. Esta configuração é particularmente importante, já que os códigos de integração continua usarão esta rota para a compilação de Minix.

2.2 Ambiente de desenvolvimento

Agora que já revisaram Minix, devem de preparar o ambiente de desenvolvimento com ele. Notem que trabalhar diretamente dentro de Minix pode ser muito difícil, pelas incompatibilidades dos drivers do teclado, restrições nas terminais, etc. Consequentemente, precisam uma maneira conveniente de mover arquivos entre sua máquina e a instalação de Minix. Podem utilizar *rsync* para sincronizar a pasta de Minix e a carpeta de desenvolvimento, ou podem usar *git* usando *pull* e *push* em cada um dos ambientes. Para poder executar *rsync*, devem de instalar *openssh* em Minix (ver Secção 2.1).

Depois de instalar os pacotes devem de copiar o *sshd* no espaço de seu usuário (por default *root*), e reiniciar o demônio de *sshd*

```
$ cp /usr/pkg/etc/rc.d/sshd /etc/rc.d/sshd
$ /etc/rc.d/sshd start
```

Para que o serviço arranque automaticamente têm que executar

```
$ printf 'sshd=YES\n' >> /etc/rc.conf
```

Isto faz que o serviço de *sshd* seja iniciado com o sistema. Podem fazer um teste simples. Reiniciem a máquina antes de executar o comando anterior (ou mudando o YES para NO) e vejam os serviços que são iniciados quando a máquina arranca.

Adicionalmente, devem de configurar a conexão a través do porto 22 (porto padrão) para poder conectar-se a su máquina virtual. Por exemplo, em *Qemu* utilizando o modo usuário de redes, podem agregar o parâmetro *-redir tcp:10022::22* à linha de comando ou *-net user,hostfwd=tcp::10022-:22* (segundo a wiki do Minix). Este argumento permite acessar ao porto 22 de Minix a través do porto 10022 de su máquina anfitriã. Se recomenda copiar sua chave pública de *ssh* no arquivo *.ssh/authorized_keys*, e que agreguem uma entrada no arquivo *.ssh/config* da máquina de desenvolvimento (não na máquina de Minix) que lhes permita fazer uma conexão direta a través do comando *ssh minix*. Podem utilizar o seguinte código como referencia inicial para o arquivo *.ssh/config* (supondo que usam *qemu*) na máquina anfitriã

```
Host minix
    Hostname localhost
    Port 10022
    User root
```

(Vejam a [seguinte guia](#) para a configuração das chaves *ssh*.) Se a configuração for correta e a máquina estiver iniciada deveriam poder acessar usando

```
$ ssh minix
```

ou se não usaram a configuração em *.ssh/config*

```
$ ssh -p 10022 root@localhost
```

Devem criar uma copia do código fonte na pasta */usr/src*, para isso podem usar *git* ou *rsync*. Podem revisar a [guia de git](#) na wiki de Minix.

2.3 Compilando e modificando o kernel

Agora é momento de recompilar o kernel. Antes de fazer qualquer mudança, tenha certeza que pode construir um novo kernel. Vá para a pasta */usr/src/tools* e execute *make*. Revisem as opções para ver como pode se instalar e reconstruir o kernel. Uma vez instalado o kernel, e depois de reiniciar

a máquina, devem selecionar a opção do novo kernel personalizado. Senão, executará o kernel anterior.

Se recomenda ficar num kernel estável quando transfirm arquivos de e para Minix, e ainda mais **importante quando estejam compilando**. Para regressar ao kernel estável podem executar o seguinte código (e troquem de opção no arranque)

```
$ shutdown -r now
```

Agora que podem construir e carregar um novo kernel, é momento de fazer uma mudança ao sistema operacional. Primeiro, devem fazer uma nova rama no seu repositório

```
$ git checkout -b projeto-1
```

Para as mudanças do kernel, devem de imprimir o nome de cada arquivo que é executado pelo sistema operacional. Devem de encontrar os arquivos fonte do kernel que implementam a chamada ao sistema `exec`. Depois, devem inserir uma instrução `printf` que emita executando: `arquivo`, onde `arquivo` é o nome do arquivo que se executa. Quer dizer, quando se escreve `ls` na terminal, deve de ler-se o string executando: `arquivo-ls`, onde `arquivo-ls` é a rota ao arquivo que contem o código fonte do comando `ls`. Notem que esta mudança é muito simples, mas dará as facilidades necessárias para entender o que sucede no sistema operacional e familiarizar-se com ele. Posteriormente, devem enviar suas mudanças na rama que criaram e sincronizá-la

```
$ git commit arquivo-1 arquivo-2 ... arquivo-N -m "Comentario bom que explica as mudanças"
```

Eu recomendo fortemente fazer `commits` conforme vão desenvolvendo para manter um histórico do trabalho. Além disso, tentem evitar usar um `commit -a` que envia todas as mudanças (que podem não ser significativas do trabalho feito).

Reconstruam o kernel, instalem-no, e reiniciem o sistema. Se tiveram sucesso, deveriam de ver as mudanças nos comandos enquanto o sistema arranca.

3 Relatório

Devem preparar um relatório de **máximo 4 páginas** de conteúdo (quer dizer que o limite não inclui figuras e referencias) utilizando `IEEEtran.cls`.

O relatório deve conter (pelo menos)

- Resumo
- Introdução (explicando Minix e sua tecnologia)
- Desenvolver os tópicos:
 - Explicar a instalação: desafios encontrados e soluções.

Tabela 1. Avaliação do projeto.

	%
Introdução	10
Instalação	20
Mudança <code>exec</code>	30
Banner	15
Chamada do sistema	15
Conclusões	10

- Explicar como fizeram a mudança do `exec` (e as fontes que consultaram para chegar à solução).
- Explicar como cambiar o banner do kernel de Minix (“Minix 3.4.0 Copyright...”) que se mostra quando Minix arranca. Como chegaram a sua resposta?
- Explicar qual é nome da chamada do sistema número 33 (decimal) de Minix, e explicar como encontraram essa informação.

- Conclusão (explicar os pontos principais do trabalho)

Se se detecta plagio os envolvidos terão zero na disciplina, e se dará aviso às autoridades correspondentes para que se tomem as sanciones do caso.

4 Dicas

- Este projeto pode parecer muito simples, e ele é. Mas a ideia dele é que usem este tempo para familiarizar-se com o código fonte de Minix e com sua organização. Senão usam este tempo, terão problemas nos próximos projetos e falta de tempo entendendo o sistema.
- Revisem a documentação oficial de Minix para poder resolver problemas.
- Estabeleçam metas iniciais, e etapas para avançar no projeto. Por exemplo, instalar Minix, logo comunicar-se e instalar programas através de `ssh`, etc. A tarefa pode demorar mais se não são organizados no uso de seu tempo, e se ficam refazendo as mesmas coisas uma e outra vez não estão avançando.
- **Não comecem o projeto dias antes da submissão.** Planejem sua entrega e façam avanços oportunamente.

5 Avaliação

O projeto contempla a avaliação do relatório (escrita, estruturação, redação, etc.) e as tarefas feitas (prá-

tica) de maneira conjunta. Se avaliará o relatório entregado, assim como o trabalho feito no repositório (através do histórico no repositório). O detalhe da avaliação está dado na Tabela 1.

Neste projeto o trabalho no git não será ponderado dada a simpleza do trabalho. Mas nos próximos projetos o trabalho de cada integrante nele terá uma contribuição importante.