

Debugando pintOS no vsCode

Para funcionamento correto do depurador, utilizaremos a pasta do pintOS com o launch.json e devcontainer.json atualizados que encontra-se no drive, com nome: "pintos_dev_container.zip".

Antes de tudo, é preciso garantir que algumas ferramentas sejam instaladas :

1 - Vscodex/ GCC / GDB:

Execute os comandos a seguir, um de cada vez, para adicionar a chave GPG do repositório do VsCode e adicionar o repositório ao seu sistema:

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor  
> microsoft.gpg
```

```
sudo install -o root -g root -m 644 microsoft.gpg /etc/apt/trusted.gpg.d/
```

```
sudo sh -c 'echo "deb [arch=amd64]  
https://packages.microsoft.com/repos/vscode stable main" >  
/etc/apt/sources.list.d/vscode.list'
```

Em seguida, atualize a lista de pacotes e instale o VsCode com os seguintes comandos, caso já tenha o vs instalado utilize apenas a primeira linha de comando:

```
sudo apt update
```

```
sudo apt install code
```

Agora é preciso instalar o GCC para compilação e o GDB para Debug:

```
sudo apt install build-essential
```

```
sudo apt install gdb
```

2 - Docker

Primeiramente verifique se você já possui o Docker instalado no seu sistema com o seguinte comando:

```
docker --version
```

Esse comando exibirá a versão do Docker instalada, se estiver presente. Se você tiver o Docker instalado, ele mostrará a versão; caso contrário, exibirá um erro informando que o comando não foi encontrado.

Para instalar use esses 6 comandos:

```
sudo apt update
```

```
sudo apt install apt-transport-https ca-certificates curl gnupg lsb-release
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor  
-o /usr/share/keyrings/docker-archive-keyring.gpg
```

```
echo "deb [arch=amd64  
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee  
/etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt update
```

```
sudo apt install docker-ce docker-ce-cli containerd.io
```

Após instalado, é necessário colocar o usuário ao grupo Docker do Linux:

```
sudo usermod -aG docker nome_do_usuario
```

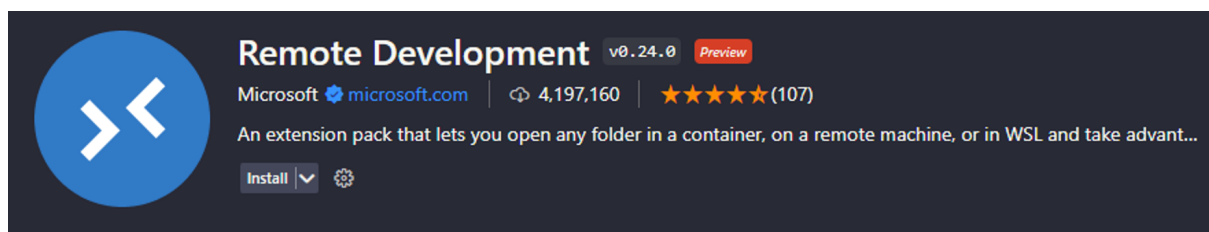
Após isso reinicie a máquina.

Para testar se está tudo funcionando digite no seu terminal:

```
docker run hello-world
```

3 - Remote Development Extension Pack

Para instalar o RD, digite 'code' no seu terminal para abrir o VsCode. Em seguida, vá em extensões e pesquise por Remote Development:



Instalando-o, ele trará consigo todas extensões necessárias para desenvolvimento usando Containers.

Criando Container para o Projeto

- Pressione **F1** no Vscode e digite **Dev Containers: Open Folder in Container;**
- Selecione a **pasta geral do projeto pintOS;**
- Selecione o **arquivo.c** que você deseja debugar;
- Marque os **breakpoints;**
- Clique em **Run and Debug;**
- Selecione **[P1] Debug a Test** como debug configuration;
- Digite qual será o nome do teste que você queira debugar.

Usamos https://github.com/LiangLouise/pintos_dev_container como referência.