

ATENÇÃO: Responda às questões **6 a 9** num conjunto de folhas e às questões **10 a 12** noutro conjunto.

6. [3] Considere o código, apresentado abaixo, de um programa cliente que comunica com um servidor que atende pedidos no *socket* de domínio Unix localizado em `/run/tp2-server.sock`. Não contando com a falta das diretivas de `#include` nem com a falta do tratamento de erros, indique os erros notórios no código apresentado, justificando devidamente porque motivo são erros.

```
struct sockaddr_un cli_addr;
int main(int argc, char * argv[]) {
    int conn_fd = socket(AF_UNIX, SOCK_STREAM, 0);
    cli_addr.sun_family = AF_UNIX;
    sprintf(cli_addr.sun_path, "/run/tp2-client-%d.sock", getpid());
    cli_addr.sun_port = 5600;
    connect(conn_fd, (struct sockaddr *)&cli_addr, sizeof (cli_addr));
    /* utilização de conn_fd */
    close(conn_fd);
    return 0;
}
```

7. [2.5] A especificação de um serviço *systemd* pode incluir opcionalmente a diretiva `ExecStop`, geralmente omissa, para especificar um comando explícito para terminar o serviço. No entanto, mesmo sem esta diretiva, o comando `systemctl stop nome_do_serviço` consegue geralmente terminá-lo. Como?
8. [2.5] Na secção de instalação do ficheiro que descreve um serviço em *systemd*, explique porque razão é preferível usar `WantedBy`, em vez de `RequiredBy`, para associar esse serviço a um *target*.
9. [3] O código fonte do *kernel* Linux pode ser compilado com diversas opções de configuração. Uma dessas opções está documentada com o seguinte texto:

«This changes the kernel so it can modify itself when it is run under a hypervisor, potentially improving performance significantly over full virtualization. However, when run without a hypervisor the kernel is theoretically slower and slightly larger.»*

in <https://github.com/torvalds/linux/blob/master/arch/x86/Kconfig>

* neste contexto significa “sobre” ou “debaixo do controlo de”

Que nome se dá ao tipo de solução de virtualização que está implícito neste texto? Na sua resposta, refira as partes relevantes do texto acima que permitem identificar o tipo de solução/otimização em causa.

10. [2.5] Descreva sucintamente 5 a 7 passos que considera relevantes na compreensão do que acontece num sistema Linux desde que se executa o comando `docker run -it ubuntu:latest` até que o respetivo contentor está em execução. NOTA: Existem múltiplas respostas válidas.
11. [2.5] Um ficheiro `docker-compose.yml`, para especificação de uma solução composta com o nome `tp2`, contém, entre outros elementos, a especificação de um serviço com o nome `svc`, associado a uma rede `tp2net` de tipo `bridge`. Ao levantar a solução com `up`, este serviço poderá ter associado mais do que 1 contentor? Se sim, como? Se não, o que poderia ser feito para que isso fosse possível? Em qualquer caso, que nome ou nomes podem ser usados na rede `tp2net` para chegar a esse(s) contentor(es)?

(continua)

12. [4] Considere o Dockerfile apresentado ao lado, os ficheiros `package.json` e `app.js`, com uma aplicação para Node.js, o ficheiro `README.md`, e uma pasta `images` com ficheiros de suporte à interface de utilizador da aplicação. Pretende-se que no sistema de ficheiros das instâncias da imagem todos os ficheiros e pastas abaixo de `/home/node/webapp` tenham como dono o utilizador `node`. Considera-se que há grande probabilidade de alterar o ficheiro `app.js`, o porto da aplicação e os ficheiros da pasta `images`. Pretende-se ainda, prioritariamente, otimizar a reutilização da *cache* de imagens intermédias e, dentro do possível, minimizar o número de camadas de *overlay*.

```
FROM node:alpine
WORKDIR /home/node/webapp
COPY README.md .
COPY images/* ./images
USER node
ENV PORT=80
COPY package.json app.js .
RUN chown -R node.node /home/node
RUN npm install
EXPOSE $PORT
CMD ["node", "app.js"]
```

- Identifique problemas no Dockerfile que contrariam os objetivos enunciados atrás.
- Corrija o Dockerfile de modo a concretizar os objetivos enunciados.

Duração: 1 hora e 15 minutos

ISEL, 2 de fevereiro de 2023