SSC141

SISTEMAS OPERACIONAIS II

Projeto I

# Funcionamento de um Teclado em Sistemas Linux e Proposta de Implemetação de um Keylogger

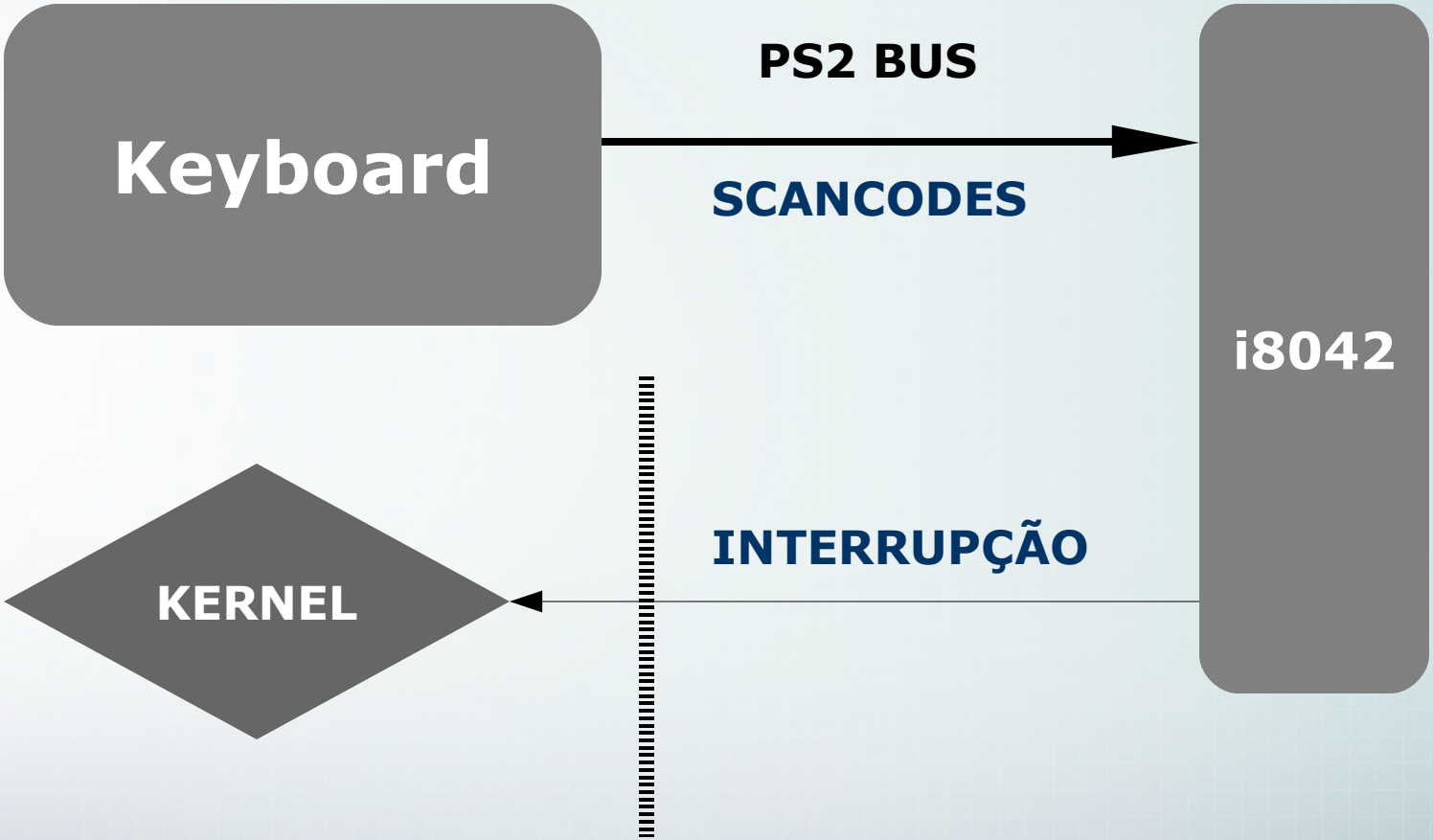

**Trabalho Desenvolvido pelos Alunos (Grupo 8A):**

**Ubiratan Soares**
**Ulisses Soares**
**Vinicius Grippa**

# Agenda

1. Como funciona um teclado com o Linux
2. Abordagens para Keylogging
3. Proposta de Implementação
4. Referências

# Teclado e Interrupção



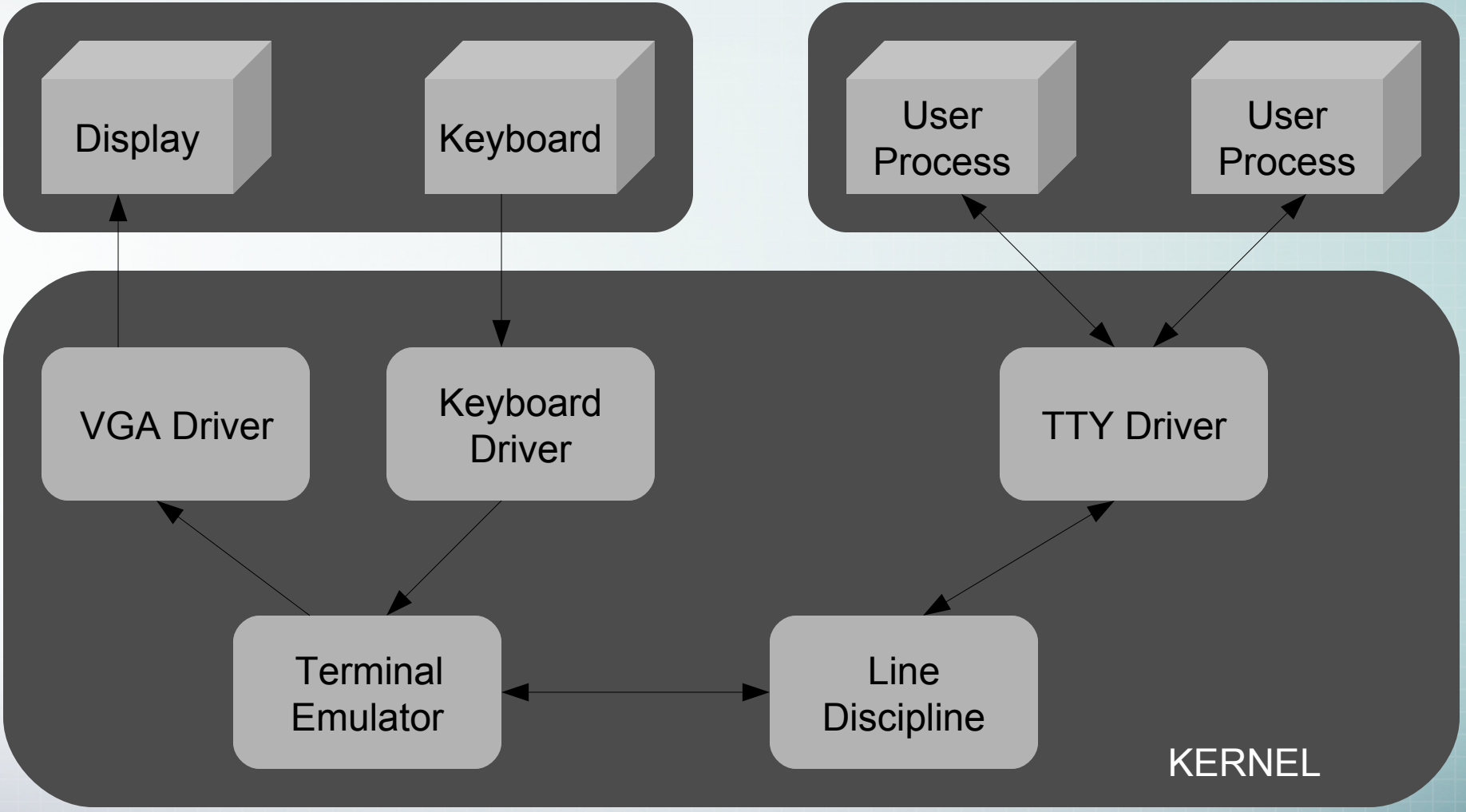
# Scancodes

- **Modos 1 a 3 , dependendo do pressionar e liberar de teclas;**
- **Usualmente, cada toque gera um *make code* e um *break code***
- **Exemplo: tecla c do teclado**
  - Make Code : **e0 c**
  - Break Code : **e0 c+0x80**
- **Sequência de até 6 scancodes por toque;**
  - Pause : **e1 1d 45 e1 9d 65**
- **Tratados pela função handle_scancode() no Kernel 2.4;**

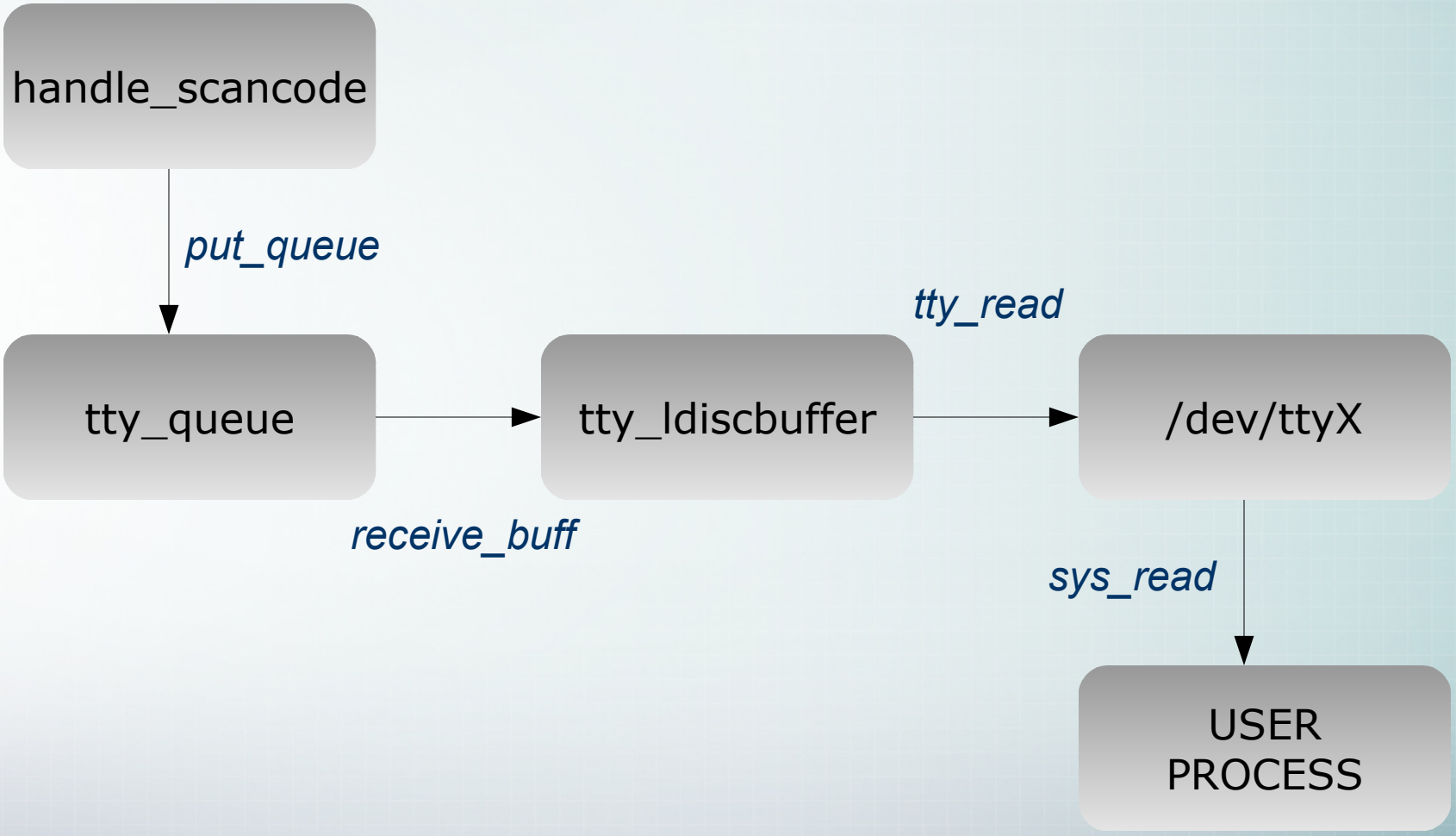
# Key Codes e Keymap

- *Parser* dos Scancodes gera um keycode;
- Cada keycode é único para cada tecla;
  - Exemplo : Tecla ESC | Keycode = 1
- O keycode é convertido em um caractere (ASCII, UNICODE) após mapeamento do layout do teclado via Keymap;
- Comando **loadkeys**;

# Caracetere, TTY e Processos



# Funções e System Calls (Kernel 2.4)



# Keylogging no Kernel

- **Modificar o Kernel**
  - */drivers/input/serio/i8042.c*
  - */drivers/input/keyboard/atkbd.c*
  - */drivers/char/keyboard.c*

  **de maneira a capturar o toque;**
- **Exemplo : *printk()*; (?)**
- **Interceptar alguma system call no caminho do caractere ao processo do usuário (mais difícil);**

# Exemplos de Kernel Keyloggers

- **Vlogger (“outdated”);**
- **TTYRPLD ;**
  - Meta Kernel (Linux, BDS`s e Solaris);
  - Suporte a vários tipos de TTY;
  - Networking;

# Kernel Keylogging – Análise

- **Vantagens**
  - Versatilidade (scancodes, keymaps);
  - Fácil User Keylogging (Telnet, SSH, etc);
  - Baixo overhead;
- **Desvantagens**
  - Codificação (File System, Rede);
  - Recompilação Necessária;
  - Depedência da versão do Kernel;
  - Abordagem pode ser dependente de arquitetura;

# Keylogging no Espaço do Usuário

- Processo do usuário acessa o hardware com permissão de root;
- Exemplo : ler diretamente a porta PS2 (0x60), USB, Wireless, etc;
- Exemplos
  - uberkey;
  - lkl (?)
  - lynspy2 (?);
  - klogger;

# User-space Keylogging - Análise

- **Vantagens**
  - Independência entre sistemas Unix-Like;
  - Acesso ao sistema de arquivos e rede facilitado;
  - Não é necessário recompilar;
- **Desvantagens**
  - *Parser* "manual"de scancodes;
  - É necessário conhecer o layout do usuário;
  - Interferência de outros dispositivos;
  - Overhead (?);

# Proposta de Implementação

- Aprimoramento do Klogger;
- Keylogger hooker no espaço do usuário, lendo diretamente a porta PS2 do hardware;
- Estudo de viabilidade de meta-keylogging, com extensão para teclado USB/ Wireless;
- Abordagem portável entre diferentes sistemas Unix-like;

# Referências

1. "*Keystroke logging*". Wikipedia, The Free Encyclopedia

   http://en.wikipedia.org/wiki/Keylogging

2. "*Linux and the Keyboard*"

   http://gunnarwrobel.de/wiki/Linux-and-the-keyboard.html

3. "*The Linux Keyboard Driver*".
   http://www.linuxjournal.com/article/1080

4. "*The TTY Demystified*" .

   http://www.linusakesson.net/programming/tty/index.php

5. "*Study of Buffer Overflows and Keyloggers in the Linux Operating Systems*".

   Carrol, Patrick - University of Baltimore, Novembro de 2007