

Trabajo Practico 1: Trabajando con Xinu

(Respuestas [aqui](#))

Objetivos

- ➡ Modificar, compilar y testear el sistema operativo **Xinu**.
- Crear y finalizar procesos (en **Xinu** y **Linux**).
- Conocer los componentes de la **tabla de procesos** en Xinu.

Entorno de Trabajo

- Usaremos **Xinu** para arquitectura **PC (x86)**.
 - Emulación mediante **QEMU** (máquina virtual básica).
 - **Recomendado:** Usar máquinas del laboratorio.
 - **Alternativa:** Instalar en Linux local (si se tiene).
-

Ejercicio 1: Primeros Pasos con Xinu

Descarga y Compilación

```
# Descargar el código fuente
wget http://se.fi.uncoma.edu.ar/so/misc/xinu-pc.tar.gz

# Desempaquetar
tar xvf xinu-pc.tar.gz

# Compilar
cd xinu-pc/compile/
make clean
make

# Ejecutar en QEMU
make run-qemu

# Acceder al shell (CONSOLA)
CTRL + ALT + 3
```

? Preguntas

1. ¿Qué componentes trae esta versión de Xinu?
 2. ¿Qué periféricos de PC son accesibles desde QEMU?
 3. ¿Cómo se accede al puerto serie en QEMU para usar el shell?
 4. ¿Cuántos procesos hay en ejecución? ¿Cómo lo obtuvo?
-

📄 Ejercicio 2: Modificación de Xinu

- 📄 Editar `main.c` para mostrar un mensaje en la **pantalla VGA a colores**.
 - 📄 **Compilar y verificar** los cambios.
-

📄 Ejercicio 3: Incorporar un Programa al Shell de Xinu

1. Escribir un **hello world** en un archivo `.c` dentro de `shell/`.
 - **¡Atención!** En Xinu, la función principal **no** se llama `main()`.

```
#include <xinu.h>
mi_programa() {
    printf("Hola mundo! \n");
}
```

2. Agregar el programa a `shell/cmdtab.c` y `include/shprototypes.h`.
 3. **Compilar y verificar** que el programa esté disponible en el shell.
 4. ¿Por qué en Xinu no se usa `main()`?
-

⚙️ Ejercicio 4: Creación de Procesos en Xinu

- 📄 Incluir código para crear **dos procesos** (ejemplo: `ej1.c`).

? Preguntas(1)

- ¿Qué sucede al ejecutar el programa?
 - ¿Cómo lograr que el sistema conmute más seguido entre procesos?
 - Investigar:
 - ¿Qué es el **QUANTUM** en SO?
 - ¿Cómo se modifica en Xinu?
-

📄 Ejercicio 5: Finalización de Procesos en Xinu

- 📄 **Modificar ejercicio anterior** para:

1. Esperar **10 segundos**.
 2. Finalizar los procesos creados anteriormente.
 - **Ayuda:** Guardar los **PIDs** y buscar la syscall para finalizar procesos.
-

📄 Ejercicio 6: Varios Procesos Reutilizando Código

- 📄 **Crear dos procesos** (ejemplo: `ej2.c`).
- 📄 **Modificar el tamaño de pila a 8KB** para cada proceso.

□ Ejercicio 7: Creación de Procesos en Linux

□ Programa en Linux que:

1. Crea un proceso hijo (calcula primos del 1000 al 5000).
2. El padre calcula primos del 0 al 1000.
3. Finaliza el hijo y luego termina con `exit()`.

? Preguntas(2)

- ¿Cuál fue el último primo calculado por el hijo?
- ¿Pudo calcular todos los primos requeridos?

□ Ejercicio 7-bis: Reemplazo de Imagen en Linux

□ Crear un proceso hijo que reemplace su imagen por `/usr/bin/date`.

□ El padre finaliza con `exit()`.

? Pregunta

- ¿Qué funciones de C realizaron llamadas al sistema?

□ Ejercicio 8: Xinu vs. Unix/Linux

□ Douglas Comer (creador de Xinu) dice:

"Xinu Is Not Unix. Xinu es más pequeño, elegante y fácil de entender."

? Preguntas(3)

1. Comparar funciones de creación de procesos en Linux y Xinu.
2. ¿Cuál es más fácil?
3. Si eliges Linux, ¿por qué Comer diría que Xinu es más elegante?

□ Ejercicio 9: Tabla de Procesos en Xinu

□ Analizar `include/process.h`.

? Preguntas(4)

1. ¿Qué campos tiene la tabla de procesos?
2. ¿Falta algo importante para gestionar procesos?

□ Ejercicio 10: Portar el Juego del Ahorcado a Xinu

□ Adaptar el juego para ejecutarse en Xinu.

□ Agregar acceso desde el shell.

? Pregunta(5)

- ¿Qué modificaciones fueron necesarias?

□ Ayudas

- **system()** no existe en Xinu → Buscar equivalente.
 - **Otras funciones de biblioteca** → Ver si hay equivalentes en Xinu.
-

□ ¡Manos a la obra! □