

U.B.A. - Facultad de Ingeniería

66.20/86.37 Organización de Computadoras

Introducción

Práctica jueves

1^{er} cuatrimestre 2019

Docentes

- ▶ Dr. Ing. Juan Heguiabehere
jheguia@gmail.com
- ▶ Ing. Tomás Niño Kehoe
tomasninokehoe@gmail.com
- ▶ Ing. Matias Stahl
stahlmatias@gmail.com

Temas

- ▶ Desempeño - Ley de Amdahl
- ▶ ISA MIPS
- ▶ Jerarquía de memorias
- ▶ Pipeline
- ▶ Datapath

Evaluación

- ▶ Parcial con dos recuperatorios
- ▶ Trabajos práctico grupal obligatorios
- ▶ Participación en clase

Herramientas

- ▶ Compilador: GCC
- ▶ Sistema de documentación: \LaTeX
- ▶ Emulador: QEMU
- ▶ Sistema de emulación gráfica MIPS: DrMIPS
- ▶ Sistema operativo host: Ubuntu 18.04.2 LTS
- ▶ Sistema operativo guest: Debian 4.9.130-2 (2018-10-27) mips

Herramientas: GCC

- ▶ Compilador C (entre otros)
- ▶ Gratuito y open source
- ▶ Soporta múltiples arquitecturas (inclusive MIPS)
- ▶ Genera código assembly

Herramientas: GCC

Supongamos que `myprog.c` es el código fuente en C a compilar:

```
$ gcc -Wall -o myexec myprog.c
```

Donde:

- ▶ `-Wall`: activa todos los mensajes de warning
- ▶ `-o`: archivo de salida (en este caso, `myexec`)

Herramientas: GCC

Para detener al compilador justo después de generar el código assembly:

```
$ gcc -Wall -O0 -S myprog.c
```

Donde:

- ▶ -S: detiene al compilador luego de generar el assembly
- ▶ -O0: No aplica optimizaciones

Esto genera el archivo myprog.s con el assembly que gcc genera para myprog.c

Herramientas: L^AT_EX

- ▶ Permite concentrarse en el contenido del documento en lugar de la forma del mismo
- ▶ Formato abierto y de texto (se pueden mantener los documentos con CVS o GIT)
- ▶ Resultados muy profesionales
- ▶ Templates tipo “paper”

Documentación

- ▶ “The Not So Short Introduction To LaTeX”

<http://tug.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

Herramientas: QEMU

Es un proyecto open source que permite emular un procesador completo incluyendo MIPS.

<https://www.qemu.org/>

Herramientas: QEMU - Preparación del entorno

Este entorno se preparará bajo el sistema operativo **host** Ubuntu 18.04.2 LTS. La única interacción con el sistema **guest** es durante la instalación de Debian MIPS.

- ▶ Instalar QEMU

```
$ sudo apt-get install qemu-system-mips
```

- ▶ Descargar Debian para MIPS-32 (Big Endian) **Malta**

```
$ wget http://ftp.debian.org/debian/dists/stable/main  
/installer-mips/current/images/malta/netboot/  
initrd.gz
```

```
$ wget http://ftp.debian.org/debian/dists/stable/main  
/installer-mips/current/images/malta/netboot/  
vmlinux-4.9.0-8-4kc-malta
```

Herramientas: QEMU - Preparación del entorno

- ▶ Crear un archivo imagen en QEMU

```
$ qemu-img create -f qcow2 hda.img 2G
```

- ▶ Instalar Debian MIPS

```
$ qemu-system-mips -M malta -m 256 -hda hda.img \  
-kernel vmlinux-4.9.0-8-4kc-malta -initrd initrd.gz \  
-append "console=ttyS0 nokaslr" -nographic
```

- ▶ Instalar el servidor SSH

En las opciones del instalador de Debian tildar la opción SSH Server

Herramientas: QEMU - Preparación del entorno

- ▶ Finalizar la instalación de Debian MIPS

En la pantalla de instalación seleccionar Go Back Luego seleccionar la opción Execute Shell y tipear el comando **poweroff**

- ▶ Copiar la imagen del Kernel (ejecutados en el **host**)

```
$ sudo modprobe nbd max_part=63
$ sudo qemu-nbd -c /dev/nbd0 hda.img
$ sudo mount /dev/nbd0p1 /mnt
$ cp -r /mnt/boot/initrd.img-4.9.0-8-4kc-malta .
$ sudo umount /mnt
$ sudo qemu-nbd -d /dev/nbd0
```

Herramientas: QEMU - Preparación del entorno

- ▶ Ejecutar la imagen

```
$ qemu-system-mips -M malta -m 256 -hda hda.img \  
-kernel vmlinux-4.9.0-8-4kc-malta \  
-initrd initrd.img-4.9.0-8-4kc-malta \  
-append "root=/dev/sda1 console=ttyS0 nokaslr" \  
-nographic -device e1000-82545em,netdev=user.0 \  
-netdev user,id=user.0,hostfwd=tcp::5555-:22
```

- ▶ Para acceder a la máquina guest desde el host

```
$ ssh root@localhost -p 5555
```

- ▶ Copiar archivos

```
$ scp -P 5555 file.txt root@localhost:/tmp
```

Herramientas: QEMU - Preparación del entorno

```
mstahl@907242N: ~  
File Edit View Search Terminal Help  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
root@debmips:~#  
root@debmips:~# cat /proc/cpuinfo  
system type      : MIPS Malta  
machine          : mti,malta  
processor        : 0  
cpu model        : MIPS 24Kc V0.0  FPU V0.0  
BogoMIPS         : 1001.47  
wait instruction : yes  
microsecond timers : yes  
tlb_entries      : 16  
extra interrupt vector : yes  
hardware watchpoint : yes, count: 1, address/irw mask: [0xff8]  
isa              : mips1 mips2 mips32r1 mips32r2  
ASEs implemented : mips16  
shadow register sets : 1  
kscratch registers : 0  
package          : 0  
core             : 0  
VCED exceptions   : not available  
VCEI exceptions   : not available  
root@debmips:~#
```

Herramientas: QEMU - Automatizando el despliegue

Se proporcionan dos scripts en bash para desplegar el entorno.

- ▶ `install_env.sh`

Instala en caso de ser necesario `wget` y `qemu-system-mips`. Descarga Debian MIPS. Ejecuta el instalador de Debian, este proceso es manual, se tienen que seguir los pasos presentados en pantalla. Una vez finalizada la instalación seleccionar Go Back. Luego seleccionar la opción Execute Shell y tipear el comando **poweroff**.

- ▶ `exec_env.sh`

Ejecuta Debian MIPS dentro de QEMU.

Herramientas: QEMU - Imagen preconstruida

Se puede probar varias arquitecturas de MIPS utilizando una imagen de disco preconstruida con su imagen del kernel. En la máquina Host se ejecuta:

```
wget https://people.debian.org/~jcowgill/qemu-mips/  
debian-stretch-mips.qcow2  
wget https://people.debian.org/~jcowgill/qemu-mips/  
initrd.img-4.9.0-4-5kc-malta.mips.stretch  
wget https://people.debian.org/~jcowgill/qemu-mips/  
vmlinux-4.9.0-4-5kc-malta.mips.stretch
```

Herramientas: QEMU - Imagen preconstruida

Una vez finalizada la descarga de los archivos necesarios, ejecutar en el sistema Host lo siguiente:

```
qemu-system-mips64 \  
  -M malta \  
  -cpu MIPS64R2-generic \  
  -m 2G \  
  -append 'root=/dev/vda console=ttyS0 mem=2048m net.  
    ifnames=0 nokaslr' \  
  -netdev user,id=user.0 \  
  -device virtio-net,netdev=user.0 \  
  -device usb-kbd \  
  -device usb-tablet \  
  -kernel vmlinux-* \  
  -initrd initrd.img-* \  
  -drive file=$(echo debian-*.qcow2),if=virtio -nographic
```

Loguearse en el sistema Guest con el usuario root (sin password).

Herramientas: QEMU - Instalar herramientas

Luego de iniciar el sistema Guest, ejecutar los siguientes comandos

```
# apt-get update
# apt-get install gcc
# apt-get install gcc
# apt-get install vim
```

Herramientas: QEMU - Hola mundo

holamundo.c

```
#include <unistd.h>
extern size_t mystrlen(const char *);

int main(int argc, char * const argv[]){
    char *msg = "Hola mundo.\n";
    write(1, msg, mystrlen(msg));
    return 0;
}
```

Herramientas: QEMU - Hola mundo

mystrlen.S

```
#include <sys/regdef.h>

.text
.align 2
.globl mystrlen
.ent mystrlen

mystrlen:
    .frame fp, 16, ra
    .set noreorder
    .cpload t9
    .set reorder

    subu sp, sp, 16
    .cprestore 0

    sw fp, 4(sp)
    move fp, sp
    li v0, 0

mystrlen_loop:
    lb t0, 0(a0)
    beqz t0, mystrlen_return
    addiu a0, a0, 1
    addiu v0, v0, 1
    j mystrlen_loop

mystrlen_return:
    lw fp, 4(sp)
    addu sp, sp, 16
    j ra
.end mystrlen
```

Herramientas: QEMU - Hola mundo

Compilar y ejecutar en el entorno guest Debian MIPS.

```
# gcc -Wall -g -o holamundo holamundo.c mystrlen.S
# ./holamundo
Hola mundo.
#
```

Links

- ▶ Grupo Yahoo
<https://groups.yahoo.com/neo/groups/orga6620>
- ▶ Grupo Slack
<https://orga6620.slack.com>

Bibliografía

- ▶ David Patterson, John Hennessy, *Computer Architecture a Quantitative Approach*, Elsevier, 3rd edition. ISBN: 1-55860-596-7. May 2002.
- ▶ David Patterson, John Hennessy, *Computer Organization and Design, the Hardware/Software Interface*, Elsevier, 3rd edition. ISBN: 1-55860-604-1. Aug. 2004.
- ▶ B.L. Jacob and T.N. Mudge, *Virtual Memory: Issues of Implementation*, Computer, Vol. 31, No. 6, June 1998, pp. 33-43.
- ▶ B.L. Jacob and T.N. Mudge, *Virtual Memory in Contemporary Microprocessors*, IEEE Micro, Aug. 1998.

Bibliografía

- ▶ Ulrich Dreper, *What every programmer should know about memory*
- ▶ Jean-Loup Baer, *Microprocessor Architecture. From Simple Pipelines to Chip Multiprocessors*, Cambridge University Press. ISBN-13 978-0-521-76992-1. 2010
- ▶ Rajeev Balasubramonian and Norman P. Jouppi and Naveen Muralimanohar, *Multi-Core Cache Hierarchies*, Morgan and Claypool Publishers, 2011.
- ▶ System V Application Binary Interface, MIPS RISC Processor, 3rd Edition, The Santa Cruz Operation, February 1996 (<http://www.sco.com/developers/devspecs/mipsabi.pdf>).