

Notas para la práctica 1

Sistemas Operativos 2

Licenciatura en Ciencias de la Computación 2024

1. Introducción

1.1. ¿Qué es Nachos?

Nachos (siglas del inglés “Not Another Completely Heuristic Operating System”) es un sistema operativo educativo para estudiantes de cursos de Sistemas Operativos en carreras de grado. Escrito originalmente en C++ para la arquitectura MIPS, Nachos se ejecuta como un proceso de usuario en el sistema operativo anfitrión. Un simulador de MIPS ejecuta el código para cualquier programa de usuario que se ejecute sobre el sistema operativo Nachos.

Fue desarrollado originalmente en la Universidad de California en Berkeley por Wayne A. Christopher, Steven J. Procter y Thomas E. Anderson entre 1991 y 1992, y lo usan numerosos centros de enseñanza. La versión que usamos en esta materia incorpora unos agregados hechos en 2007 por José Miguel Santos Espino, de la Universidad de Las Palmas de Gran Canaria. Desde 2015, además, los docentes de esta materia venimos realizando diversos cambios, adaptaciones y agregados para tener una base más moderna, potente y acorde a nuestras necesidades.

Recursos:

- Repositorio de Nachos usado en la cátedra
- Sitio web oficial de Nachos
- Hoja de ruta de Nachos

1.2. ¿Cómo obtener Nachos?

La versión de Nachos que usamos en la cátedra se encuentra disponible en el repositorio Subversion del que disponemos en la carrera. En el directorio **Public** de la materia hay un subdirectorio **nachos**. Dentro del mismo, tenemos subdirectorios por años, con las distintas versiones de Nachos que hemos ido usando para el dictado de cada año.

Dentro de 2024, el subdirectorio **nachos-unr24a** contiene todo el árbol de archivos de Nachos.

Como paso inicial para trabajar con Nachos, cada alumno o grupo debe hacerse una copia del directorio **nachos/2024/nachos-unr24a**. La sección que sigue brinda más información al respecto.

1.3. Preparando el entorno de trabajo con subversion

Antes que nada, se recuerda que está disponible la guía de Subversion provista en Arquitectura del Computador para referencia de comandos y de los aspectos generales del trabajo con Subversion.

Todas las resoluciones de las planchas (excepto la plancha 1) deben subirse al repositorio Subversion de la carrera. El directorio correspondiente a Sistemas Operativos II es R-412. Dentro del mismo, hay un subdirectorio **Alumnos**, con subdirectorios para cada año.

La ruta completa es:

`https://svn.dcc.fceia.unr.edu.ar/svn-no-anon/lcc/R-412/Alumnos/2024/`

Notas:

- El repositorio svn puede utilizarse en forma de sólo lectura (para los contenidos públicos) sin necesidad de autenticarse, en ese caso se debe quitar **-no-anon** de la URL mencionada, por ejemplo: `https://svn.dcc.fceia.unr.edu.ar/svn/lcc/R-412/Public/` permite acceder al repositorio público de la materia.
- El repositorio también puede visualizarse en un navegador de internet. Si se incluye **-no-anon** en la URL entonces el navegador solicitará usuario y contraseña y permitirá acceder a los contenidos autorizados para ese usuario (incluyendo los contenidos públicos). Si no se incluye **no-anon** sólo permitirá acceder a los contenidos públicos. En cualquiera de los dos casos no se pueden hacer modificaciones al contenido.

Cada alumno o grupo debe crearse un directorio dentro de R-412/Alumnos/2024. El nombre del directorio debe incluir los apellidos de cada integrante. Dentro de este, el contenido se organizará siguiendo una estructura de subdirectorios determinada: un directorio **trunk**, que contendrá el árbol de Nachos donde se hará el trabajo continuo a lo largo del cursado, y otro **tags**, donde se copiará el contenido de **trunk** para las entregas de cada práctica.

Con **svn checkout**, **svn mkdir** y **svn commit** se puede preparar esta jerarquía, por ejemplo (cambie **eruiz** y **feroldi.ruiz.street** por su usuario e integrantes del grupo respectivamente) :

```
$ svn co https://svn.dcc.fceia.unr.edu.ar/svn-no-anon/lcc/R-412/Alumnos/2024 \
--username eruiz
$ svn mkdir feroldi_ruiz_street
$ cd feroldi_ruiz_street/
$ svn mkdir trunk
$ cd trunk
$ svn copy \
https://svn.dcc.fceia.unr.edu.ar/svn-no-anon/lcc/R-412/Public/nachos/2024/\
nachos-unr23a/code/ .
$ cd ..
$ svn mkdir tags
$ svn commit .
```

Notas:

- Recuerde que svn gestiona las diferencias de los archivos, por eso es muy importante que su copia de trabajo sea una copia generada con **svn copy** en vez de una copia normal de archivos (i.e.: a traves de un explorador o el comando **cp**).
- La barra invertida (“\”) indica que el comando continúa en la siguiente línea. No es necesario utilizarla si coloca todo el comando en la misma línea. Si se utiliza, no debe haber ningún espacio adicional al final de la línea.
- La primera vez que utilice svn en una computadora es necesario especificar el usuario con **--username** para poder autenticarse. Luego el usuario y contraseña quedan almacenados y no es necesario volver a utilizarlos.
- En máquinas compartidas con varios usuarios (particularmente en los laboratorios de la facultad) conviene agregar la opción **--no-auth-cache** a los comandos de svn. Esa opción provoca que no se almacenen el usuario y contraseña. Ej:

```
$ svn co https://svn.dcc.fceia.unr.edu.ar/svn-no-anon/lcc/R-412/Alumnos/2024 \
--username eruiz --no-auth-cache
```

1.4. Trabajo diario con subversion

Mientras se está resolviendo una plancha se debe trabajar en el directorio **trunk**, agregando los cambios cuando lo considere conveniente con **svn add** (para archivo nuevos) y **svn commit**. Recomendaciones para el commit:

- Antes de hacer un commit realice **make clean** para borrar archivos innecesarios y **svn status** para verificar los cambios. También conviene utilizar **svn diff** para verificar los cambios dentro de un archivo.
- Evite agregar archivos innecesarios al repositorio (en particular binarios como todos los archivos con sufijo **.o** y **nachos**).
- Si utiliza el repositorio desde distintas máquinas utilice **svn update** para mantener las copias locales actualizadas con el repositorio.

1.5. Entrega de una plancha

Antes de realizar una entrega verifique con **svn status** que todo lo que está en el **trunk** está “commiteado” en el repositorio. Luego cree una etiqueta de la siguiente manera (cambie **plancha2** por el número que corresponda):

```
$ svn mkdir tags/plancha2
$ svn copy trunk/* tags/plancha2
$ svn commit tags/plancha2
```

1.6. Correcciones a una entrega

Nunca modifique el contenido de una etiqueta, si es necesario realizar alguna corrección posteriormente a una entrega realice una nueva copia:

```
$ svn mkdir tags/plancha2-entrega2
$ svn copy trunk/* tags/plancha2-entrega2
$ svn commit tags/plancha2-entrega2
```

1.7. Búsquedas en el árbol de archivos

Al familiarizarse con código que escribió otra persona, frecuentemente hace falta localizar el archivo donde está cierta definición. Esto se agiliza mucho usando herramientas adecuadas.

En cualquier sistema Unix, el comando **grep** es de ayuda para esta tarea. Se invoca con la sintaxis:

```
$ grep <patrón> <archivo>...
```

También se puede invocar de forma recursiva sobre el directorio actual:

```
$ grep -r <patrón>
```

En algunos sistemas existe **rgrep**, definido como un alias de **grep -r**.

Para más información, consulte la página de manual, **grep(1)**:

```
$ man 1 grep
```

El manual en formato GNU Info, en caso de estar instalado, es una referencia extensa:

```
$ info grep
```

Existen alternativas más recientes a **grep** que mejoran diversos aspectos, si bien no son estándares ni ubicuas. También hay herramientas de etiquetado de identificadores (como **ctags** y **etags**), editores de texto y entornos de desarrollo integrados que incorporan sus propias funcionalidades de búsqueda equivalentes. Animamos a los alumnos a que investiguen estas herramientas y encuentren aquellas que mejor se integren con su forma de trabajo.