

# Laboratorio reti AA 2008/2009

**Dott. Matteo Roffilli**

**roffilli@csr.unibo.it**

**Da oggi ricevimento in ufficio**

**Previa richiesta via email**

# Laboratorio reti AA 2008/2009

**Per esercitarvi fate SSH su:**

`alfa.csr.unibo.it`

`si-tux00.csr.unibo.it`

`....`

`si-tux15.csr.unibo.it`

**Eventuali variazioni di orario/giorno verranno comunicate in anticipo via mail.**

# Laboratorio reti AA 2008/2009

- **Marzo**
  - 5 Intro,SSH,VI/VIM,GCC base
  - 12 Richiami di C e Compilazione
  - 19 Socket e Co.
  - 26 Socket e Co. parte seconda
- **Aprile**
  - 2 Utilizzo di IPC: Client
  - 16 Client parte seconda
  - 23 Server
  - 29 Server parte seconda e comandi windows
- **Maggio**
  - **21 Esercitazione e comandi linux**
  - 28 Simulazione **PROVA ESAME**

# Esercizio extra III

## Goal:

- Rendere il server multi-processo con `fork ( )`  
o multi-thread con `lpthread`

## Requisiti:

1. Accettare e gestire connessioni multiple

## Tempo a disposizione:

60 minuti

# fork()

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

main()
{
    int pid; //pid_t pid;

    pid = fork();

    if(pid == 0)
    {
        /* Processo figlio.
         * Se fork() ritorna 0, siamo all'interno
         * del processo figlio
         * Conteremo fino a dieci, una volta al secondo.
         */
        int j;
        for(j=0; j < 10; j++)
        {
            printf("Figlio: %d\n", j);
            sleep(1);
        }
        _exit(0); /* Notare che non viene usata la exit() */
    }
    else if(pid > 0)
    {
        /* Processo padre, la fork ha restituito
         * il pid del figlio.
         * Contiamo ancora fino a dieci.
         */
        int i;
        for(i=0; i < 10; i++)
        {
            printf("Padre: %d\n", i);
            sleep(1);
        }
    }
    else
    {
        /* Errore. */
        fprintf(stderr, "Errore nel fork");
        exit(1);
    }
}
```

# Utilità di sistema Linux

**ping**

**traceroute**

**netstat**

**nmap**

**netcat**

**nslookup** (per interrogare il DNS)

**ifconfig** (solo come root)

**route** (solo come root)