

# Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

29 Giugno 2007

1. (a) La funzione `f2` riceve un argomento intero e restituisce il maggiore tra il suo argomento e un numero generato casualmente. La funzione `f1(int a, int b)` scrive sullo standard output una matrice di `a` righe e `b` colonne, in cui l'elemento  $(i, j)$  è  $\max\{i \times j, \text{rand}()\}$  (risultato di `f2(i * j)`). Infine, il programma principale riceve due argomenti da riga di comando e interpreta il primo come numero di righe e il secondo come numero di colonne della matrice da stampare.
- (b) Una possibile traduzione è la seguente:

```
.text                                imull %edi
.global f1                           pushl %eax
f1:  pushl %ebp                        call  f2
      movl %esp, %ebp                addl  $4, %esp
      pushal                          pushl %eax
      # uso %esi come i                call  f3
      # uso %edi come j                addl  $4, %esp
      # 8(%ebp) e' a                   incl  %edi
      # 12(%ebp) e' b                  jmp   ciclo2
                                       avanti2:call f4
                                       incl  %esi
ciclo1:  cmpl 8(%ebp), %esi            jmp   ciclo1
      jge  avanti1                    avanti1:popal
      movl $0, %edi                   leave
ciclo2:  cmpl 12(%ebp), %edi          ret
      jge  avanti2
      movl %esi, %eax
```

2. (a) `#include <stdio.h>`  
`#include <fcntl.h>`
- `#define MAX_LINE 1024`
- ```
int main(int argc, char* argv[])
{
    FILE *f;
    char c;
    char buf[MAX_LINE], *ptr = NULL;

    if (argc != 2) {
        fprintf(stderr, "Uso: %s <file>", argv[0]);
        exit(1);
    }

    if ( (f = fopen(argv[1], "r")) == NULL) {
```

```

        perror(argv[1]);
        exit(1);
    }

    while (read(0, &c, 1) > 0) {
        switch (c) {
            case 'r':
                if (fgets(buf, MAX_LINE, f) == NULL)
                    fprintf(stderr, "?\n");
                ptr = buf;
                break;
            case 'p':
                if (!ptr)
                    fprintf(stderr, "?\n");
                else
                    printf("%s", ptr);
                break;
            case 'd':
                if (!ptr || !*ptr)
                    fprintf(stderr, "?\n");
                else
                    ptr++;
                break;
            case '\n':
                break;
            default:
                fprintf(stderr, "?\n");
                break;
        }
    }
    return 0;
}

```

```

(b) #include <stdio.h>
#include <unistd.h>
#include <string.h>

#define MAX_FILES 10
#define MAX_CMD 80

int main(int argc, char* argv[])
{
    int fd[2];
    int p[MAX_FILES + 1];
    int i, quanti, n;
    char cmd[MAX_CMD];

    if (argc < 2) {
        fprintf(stderr, "Uso: %s <file1> <file2> ... \n", argv[0]);
        exit(1);
    }
}

```

```

}

if (argc - 1 > MAX_FILES) {
    fprintf(stderr, "Massimo %d file\n", MAX_FILES);
    exit(1);
}

for (i = 0; i < argc - 1; i++) {
    p[i + 1] = 0;
    if (pipe(fd) < 0) {
        perror(argv[0]);
        break;
    }
    switch (fork()) {
    case -1:
        perror(argv[0]);
        break;
    case 0:
        close(0);
        dup(fd[0]);
        close(fd[0]);
        close(fd[1]);
        execl("editor", "editor", argv[quanti + 1], NULL);
        perror("editor");
        exit(1);
    default:
        quanti++;
        p[i + 1] = fd[1];
        close(fd[0]);
        break;
    }
}
if (!quanti)
    exit(1);
while (scanf(" %d %s", &n, cmd) == 2) {
    if (n < 1 || n > quanti || !p[n]) {
        fprintf(stderr, "numero non valido\n");
        continue;
    }

    if (write(p[n], cmd, strlen(cmd)) < 0) {
        perror(argv[n]);
        p[n] = 0;
    }
}
for (i = 1; i <= quanti; i++)
    if (p[i])
        close(p[i]);
while (quanti) {
    wait(0);
    quanti--;
}
return 0;

```

}