

STAMPA A POLLING → 8'195+50'177+2=58'374

- `copy_from_user(buffer, p, count);` → * } 8'195
`for (k=0; k<count; ++k) {` → 1 ciclo per statment iniziale +2 cicli (SLT+BNE)
`while (*printer_status_reg != READY);` → 4(2 per lettura I/O+BNE+J)*10 (tentativi [H])
`*printer_data_register = p[k];` → 3(2 per calcolo offset+1 lettura)+ 2 (scrittura I/O) cicli
`}` → 2 cicli (ADDI+J) → 1 + (2+40+5+2)*1024=50'177 cicli
`return_to_user();` → ([C]) 2 cicli

STAMPA A INTERRUPT → 8'195+47+20'469=28'711

- Implementazione dalla system call:
`copy_from_user(buffer, p, count);` → * } 8'195
`while (*printer_status_reg != READY);` → 4(2 per lettura I/O+BNE+J)*10(tentativi)
`*printer_data_register = p[0];` → 2(scrittura I/O)+1(lettura) cicli
`count = count - 1;` → (ADDI) 1 ciclo
`enable_interrupts();` → ([C]) 2 cicli
`scheduler();` → ([E]) 3 cicli
 } 47
- Implementazione della routine di servizio: → 1023 * (3(ricon [E])+3(lanc.int [E])) + 1022*14 + 1*23=20'469
`if (count == 0) {` → (BNE) 1 ciclo
`unblock_user();` → ([D]) 15 cicli
`}` else { → (J jump) 1 ciclo
`*printer_data_register = p[k];` → 3(2 per calcolo offset+1 lettura)+ 2 (scrittura I/O) cicli } 14
`count = count - 1;` → (ADDI) 1 ciclo
`k = k + 1;` → (ADDI) 1 ciclo } 23
`}`
`acknowledge_interrupt();` → ([B]) 4cicli
`return_from_interrupt();` → ([C]) 2 cicli

```
* copy_from_buffer:
(a0=&buffer, a1=&p, a2=count)

ADD $t1, $0, $0 →1
LOOP: SLT $t0, $t1, $a2 →1
      BNE $t0, $0, FINE →1
      ADDI $t1, $t1, 1 →1
      ADD $t0, $a0, $t1 →1
      ADD $t1, $a1, $t1 →1
      LW $t2, 0($t0) →1
      SW $t2, 0($t1) →1
      J LOOP →1
FINE: → 1024*8+3 = 8'195 cicli
```

STAMPA A DMA → 8'195+23+3(ricon [E])+3(lanc.int [E])+21=8'245

- Implementazione dalla system call:
`copy_from_user(buffer, p, count);` → * } 8'195
`setup_DMA_controller();` → ([A]) 20 cicli
`scheduler();` → ([E]) 3 cicli
- Implementazione della routine di servizio:
`acknowledge_interrupt();` → ([B]) 4 cicli
`unblock_user();` → ([D]) 15 cicli
`return_from_interrupt();` → ([C]) 2 cicli