```
page_pde_offset = (__PAGE_OFFSET >> 20);

        movl $(pg0 - __PAGE_OFFSET), %edi
        movl $(swapper_pg_dir - __PAGE_OFFSET), %edx
        movl $0x007, %eax       # 0x007 = PRESENT+RW+USER
10:
        leal 0x007(%edi), %ecx  # Create PDE entry
        movl %ecx, (%edx)       # Store identity PDE entry
        movl %ecx, page_pde_offset(%edx)  # Store kernel PDE entry
        addl $4, %edx
        movl $1024, %ecx
11:
        stosl   # movl %eax, (%edi)
                # addl $4, %edi
        addl $0x1000, %eax
        loop 11b
        # End condition: we must map up to and including INIT_MAP_BEYOND_END
        # bytes beyond the end of our own page tables; the +0x007 is the
        # attribute bits
        leal (INIT_MAP_BEYOND_END + 0x007)(%edi), %ebp
        cmpl %ebp, %eax
        jb 10b
        movl %edi, (init_pg_tables_end - __PAGE_OFFSET)
```